Digital Twin-Aided Vehicular Edge Network: A Large-Scale Model Optimization by Quantum-DRL

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

***Abstract*—This paper presents an innovative large model framework for optimizing the task offloading efficiency in vehicular edge networks, with a focus on ultra-reliable low-latency communication. We introduce a comprehensive model that integrates quantum computing with a deep reinforcement learning (DRL) model, supported by long short-term memory (LSTM) networks and a digital twin framework. This integration is designed to address the complexities of distributed vehicular edge computing networks, targeting efficient latency, energy, and quality-of-service management. Our model utilizes the parallel processing capabilities of quantum computing to enhance the DRL algorithm, effectively handling high-dimensional decision spaces. LSTM networks provide predictive insights into future network states in a digital twin framework and ensure real-time synchronization and adaptive strategy optimization. We employ a multi-agent framework, encompassing vehicles, unmanned aerial vehicles, and base stations, each utilizing a Nash equilibrium-based strategy for optimal decision-making, supplemented by incentive and penalty functions for reward optimization. Simulation results demonstrate notable improvements in task offloading efficiency, highlighting the model's efficacy over conventional DRL models.**

***Index Terms*—Vehicular edge computing, task offloading, ultra-reliable low-latency communication, quantum deep reinforcement learning, large model framework, long short-term memory networks, digital twin, and multi-agent systems.**

I. Introduction

FUSION of large models (LMs) and intelligent transportation systems (ITS) marks a revolutionary development in the domain of future vehicles and transportation [1]. ITS needs sophisticated artificial intelligence-enabled LMs as we move into the era of the sixth generation (6G) networks [2]. Large language models (LLMs) can revolutionize 6G communication by analyzing data, improving signal processing, and generating code for hardware development, creating intelligent wireless networks that adapt for an enhanced user experience. A robust LMs model could encompass LLMs as a subcategory in its framework. The LMs, empowered by advanced computational technologies like quantum computing and LLMs redefine the capabilities of large-scale models [1]–[3]. Recent advancements in ultra-definition multimedia streaming and data-intensive applications emphasize the need for high-end task processing capabilities, especially for services under the constraints of ultra-reliable low-latency communication (URLLC) [4]. While modern mid/low-budget vehicles lack the specialized hardware for on-board processing of resource-intensive tasks, mobile edge computing (MEC) offers a solution by bringing computational resources closer to the data source [5]. MEC integration into vehicles is limited by space, power, and cost constraints. Therefore, MEC units are mostly installed at static locations like base stations (BS) and/or in specialized aerial vehicles [4], [6], [7]. In scenarios with high computational demands, vehicles can offload these tasks to MEC and/or to unmanned aerial vehicles (UAVs), thus optimizing network efficiency and resource management.

In this context, the introduction of digital twins (DT) in ITS represents a potential paradigm shift [4]. By creating virtual replicas of physical entities, DT enables real-time monitoring, simulation, and predictive analytics. This allows for more effective resource management and optimization in task-offloading schemes. Vehicles, facing high computational demands, can employ DTs for strategic task offloading to MEC units, either through BS and/or UAVs [4], [8]. Building on the transformative role of DT in ITS and the evolving landscape of 6G networks, the incorporation of modern machine learning techniques and artificial intelligence becomes imperative for optimizing large network models [8].

A. Motivations of the work

The authors in [9] proposed a MEC service to the vehicle by a stationary BS. However, standalone BS topologies serving vehicles, particularly in areas with heavy traffic loads, currently face challenges related to bottleneck issues. However, geographical constraints often limit the establishment of BS in certain areas. Recent studies focus on these challenges [4], [10]. A distributed task offloading strategy, covering a wide geographical area, not only addresses these bottlenecks but also enhances network coverage for vehicular edge computing (VEC) services. Modern UAVs offer a solution, providing greater flexibility and potential for direct line-of-sight (LoS)

The work of K. Singh and C.-P. Li was supported in part by the National Science and Technology Council of Taiwan under Grants NSTC 112-2221-E-110-038-MY3, NSTC 112-2811-E-110-018-MY3 and in part by Sixth Generation Communication and Sensing Research Center funded by Higher Education SPROUT Project, Ministry of Education of Taiwan. The work of O. A. Dobre was supported in part by the Canada Research Chairs Program, CRC-2022-00187. The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109. (*Corresponding author: Keshav Singh.*)

A. Paul, K. Singh, and C-P. Li are with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (Email: apaul@ieee.org, keshav.singh@mail.nsysu.edu.tw, cpli@faculty.nsysu.edu.tw).

O. A. Dobre is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL A1C 5S7, Canada (E-mail: odobre@mun.ca).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT7 1NN Belfast, U.K., and also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, South Korea (e-mail: tduong@mun.ca).

connections with vehicles. Researchers thoroughly investigated the integration of heterogeneous distributed computing models for edge service provision in modern ITS [4], [7], [10].

Additionally, the complex task of resource allocation across MEC, vehicles, UAVs, BS, and other entities in the VEC network presents a dynamic and time-variant optimization challenge. Deep reinforcement learning (DRL) emerges as an appropriate solution, effectively handling the stochastic nature of edge computing resources and channel state information (CSI) [4], [8], [11]. Zhao *et al.* explored a deep deterministic policy gradient (DDPG) based DRL algorithm to train the offloading strategy in [12]. However, another recent study proves that a proximal policy optimization (PPO)-based DRL algorithm offers numerous advantages over DDPG in terms of sample efficiency, stability, ease of implementation, adaptability to constraints, handling sparse rewards, and suitability for continuous action spaces [13]. Furthermore, DT plays a crucial role in modern wireless services, creating virtual replicas of the physical world. This capability allows for more flexible optimization decisions based on historical data and pattern analysis within the DT framework [14]. Research [4], [15] leads to the development of a DT-aided multi-agent DRL system for task offloading, which assists edge computing in vehicular networks and proves more efficient than centralized DRL techniques.

Further enhancing the capabilities of the DT-DRL task offloading model, Chen *et al.* explored long short-term memory (LSTM) networks to process and analyze temporal data effectively [16]. In a network where conditions and user demands are stochastic, LSTMs provide the foresight needed for predictive resource allocation and proactive task offloading, aligning perfectly with the predictive analytics facilitated by DTs. The promising potential of quantum computing [17] in the DRL model introduces a quantum leap in our network optimization capabilities [18]. By exploiting the immense processing power of quantum computing, our DRL framework enables faster and more accurate decision-making in high-dimensional environments typical of 6G ITS networks.

B. Contributions of the Work

This work proposes novel research, integrating advanced technological paradigms in DTs, quantum-aided multi-agent DRL, and VEC within the ITS framework to serve the URLLC services to the moving vehicles. The key contributions of our work are outlined as follows:

1) Synchronized Operation of DT and Physical World Agents: Our approach uniquely employs DTs to utilize historical data using the deep learning-based LSTM model for more informed decision-making. Using DT and physical agents for the same entity, we introduce a novel operational model. This simultaneous and parallel operation aims for real-time convergence and periodic synchronization.
2) Distributed Edge Computing in ITS Networks: Our research takes a deep dive into distributed edge computing within ITS networks. We emphasize the strategic processing of tasks within the edge network under specific constraints, ensuring the timely servicing of vehicular URLLC's latency-sensitive task offloading requests.
3) Quantum-Enhanced DRL for Vehicular Task-Offloading Services: We explore quantum computing within a multi-agent DRL framework for practical applications in vehicular task offloading. This exploration addresses the unique challenges posed by vehicular networks' dynamic nature, building upon insights from a foundational study in [18]. Through a quantum-aided multi-agent DRL framework, our work makes significant progress in managing entanglement. This intricate process is crucial for optimizing a global function while accounting for the individual dynamics of each participating agent.
4) Enhanced Task Offloading Efficiency: A major highlight of our work is the numerical analysis demonstrating that our proposed model significantly improves task offloading efficiency over the conventional DRL algorithms. This advancement is quantitatively validated, showcasing the practical efficacy of our model in optimizing network resources and performance in real-world scenarios.

The proposed Quantum-DRL framework, which includes LSTM in a DT architecture, has significant importance for vehicular networks. It can improve traffic management, safety, and the effectiveness of autonomous driving systems. The framework optimizes VEC for URLLC services and promises to improve intelligent transportation systems by enabling better real-time data exchange, dynamic traffic control, and support for autonomous vehicle decision-making processes. Its application also extends to smart city infrastructure, offering solutions for energy optimization, congestion reduction, and emergency response services. This integration enables advanced vehicular network functions and promotes sustainable urban mobility and safety.

Rest of the paper is organized as follows: Section II provides a detailed description of the system model. In Section III, the problem is formulated. Section IV explains how the present problem is transformed into a DRL framework. Section V elaborates on the use of LSTM modeling in DT, and the integration of digital and physical world using a multi-agent quantum computing-aided DRL framework. Section VI presents the numerical results and analysis, demonstrating the efficiency of our model over traditional DRL methods. Finally, in Section VII, the paper summarizes the key contributions and suggests further scope of research.

II. System Model

We consider an urban terrain where vehicles, identified as $\mathcal{V} = \{1, \ldots, V\}$, are provisioned with URLLC technology to facilitate real-time task processing within strict latency constraints. These vehicles, geared to handle substantial tasks, connect with nearby single BS and/or UAVs for efficient edge computing. Each BS denoted as $b \in \mathcal{B} = \{1, \ldots, B\}$, and UAVs, indexed by $u \in \mathcal{U} = \{1, \ldots, U\}$, are equipped with advanced MEC, competent at managing intensive tasks processing [7]. Each BS in the network is equipped with $\mathcal{M}$ antennas, where $\mathcal{M} = \{1, \ldots, M\}$. In contrast, every vehicle and UAV operates with a single antenna setup. Vehicles can

connect with the b-th BS while simultaneously offloading tasks to u-th UAV. This dual-connectivity framework is pivotal in ensuring seamless delivery of URLLC services.

In instances where a vehicle falls outside the coverage of any BS, it solely relies on UAVs for task offloading. This approach highlights the unique flexibility of the network architecture, where BSs and UAVs can interconnect with other UAVs to form a sophisticated network structure. We consider that the present system model deals with the challenge of establishing direct links between BS due to complex topographies such as hill stations. Due to geographical and infrastructural constraints, achieving direct LoS connectivity between BSs is often not feasible. Establishing wired connections between BSs can also be obstructed by various obstacles, making such an approach expensive and difficult to implement in certain environments. Instead, we rely on the mobility and flexibility of UAVs to facilitate indirect BS-to-BS connections. By positioning UAVs at optimal altitudes, we can ensure direct LoS communication links between BS, thereby cost-effectively overcoming geographical limitations. Moreover, the inherent mobility of UAVs offers significant advantages, allowing for dynamic redeployment based on traffic load and network conditions, thereby enhancing the network's adaptability and efficiency in managing URLLC services. The various permissible communication configurations in the present VEC network are illustrated in Fig. 1. This figure illustrates the comprehensive connectivity possibilities within the VEC network, highlighting the interplay between vehicles, BSs, and UAVs in the task offloading process.

A. Vehicle's Task Offloading Model

The task offloading model in the vehicular network employs an advanced method to distribute priority among subtasks. This model dynamically calculates the priority based on computational complexity, data size, and intrinsic urgency of tasks, ensuring effective management of task offloading from vehicles to BSs and/or UAVs. For each task $\mathscr{T}_i$ at time instant t from a vehicle v, which is divided into subtasks $\{\mathscr{T}_{i,1}, \mathscr{T}_{i,2}, \ldots, \mathscr{T}_{i,n}\}$, such that $\mathscr{T}_i = \{\mathscr{T}_{i,1} \cup \mathscr{T}_{i,2} \cup \cdots \cup \mathscr{T}_{i,n}\}$. The model calculates the priority $\mathscr{P}_{i,j}$ of each subtask $\mathscr{T}_{i,j}$, which is defined as:

$$\mathscr{P}_{i,j}(t) = \Big(\alpha_p(t)\frac{C_{i,j}(t)}{\sum_{k=1}^{n} C_{i,k}(t)} + \beta_p(t)\frac{D_{i,j}(t)}{\sum_{k=1}^{n} D_{i,k}(t)} + \gamma_p(t)\chi_{i,j}(t)\Big)\varpi_{i,j}(t)\varkappa_{i,j}(t), \tag{1}$$

where $C_{i,j}$ and $D_{i,j}$ denote the computational complexity to process 1 bit of data and data size of subtask $\mathscr{T}_{i,j}$, respectively. $\chi_{i,j} \in (0,1)$ stands for the intrinsic task priority. $\varpi_{i,j} \in (0,1)$ and $\varkappa_{i,j} \in (0,1)$ represent the complexity weight and data size weight for the subtask, respectively. To ensure a balanced impact of the respective components, the coefficients α_p, β_p, and γ_p are subject to the constraint:

$$\alpha_p + \beta_p + \gamma_p = 1, \quad \text{with } \alpha_p, \beta_p, \gamma_p \in (0,1). \tag{2}$$

Binary decision variables $\Gamma_{v,b}^{i,j} = \{0,1\}$ and $\Gamma_{v,u}^{i,j} = \{0,1\}$ indicate whether the subtask $\mathscr{T}_{i,j}$ of vehicle v offloads to BS $b \in \mathcal{B}$ or UAV $u \in \mathcal{U}$. The optimization function, formulated

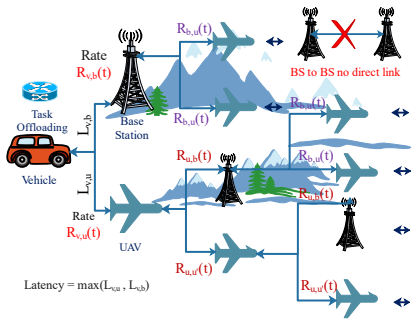

Fig. 1: Interconnected distributed edge node for faster task offloading.

to minimize the total latency across the network, incorporates this detailed priority allocation:

$$\min \sum_{v\in\mathcal{V}}\sum_{b\in\mathcal{B}}\sum_{i,j} \mathscr{P}_{i,j}(t)\Gamma_{v,b}^{i,j}(t)L_b^{i,j}(t) + \sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{U}}\sum_{i,j} \mathscr{P}_{i,j}(t)\Gamma_{v,u}^{i,j}(t)L_u^{i,j}(t), \tag{3}$$

where $L_b^{i,j}$ and $L_u^{i,j}$ are the processing latency of the offloaded subtask $\mathscr{T}_{i,j}$ through b-th BS and u-th UAV, respectively.

B. Proposed Digital Twin (DT) Model

We introduce a DT for the current vehicular network, incorporating a virtual counterpart of the physical network. This model encompasses vehicles ($\mathcal{V}$), BSs ($\mathcal{B}$), and UAVs ($\mathcal{U}$), alongside key attributes such as computational resources and connectivity status. The DT mimics each network component, featuring attributes like location, connectivity, available resources, and task load. The DT continuously synchronizes with the physical network, providing insights for real-time adjustments and strategy optimizations.

In the DT model, the CPU frequency (f_{cpu}) is pivotal in assessing the task offloading efficiency. For a base station $b \in \mathcal{B}$ or UAV $u \in \mathcal{U}$, the available MEC CPU frequency at time t is modeled as:

$$f_{\text{cpu}}(t) = f_{\max}(t) - \Delta f(\mathscr{T}_{\text{ld}}(t), \mathscr{T}_{\text{amb}}(t)) f_{\max}(t), \tag{4}$$

where $f_{\max}(t)$ denotes the maximum achievable MEC CPU frequency. The adjustment function $\Delta f(\mathscr{T}_{\text{ld}}(t), \mathscr{T}_{\text{amb}}(t))$ is defined as a complex normal distribution, $\mathcal{CN}(0, 0.1)$, accounting for variations due to load and ambient temperature. The latency for offloading a task $\mathscr{T}_{i,j}$ from various sources to destinations are defined considering different scenarios in the network:

$$v \to b: \; L_{v,b}^{i,j}(t) = \frac{D_{i,j}(t)}{R_{v,b}(t)} + \kappa_{i,j}^{b}(t)\frac{C_{i,j}(t)D_{i,j}(t)}{f_{\text{cpu}}^{b}(t)}, \tag{5a}$$

$$v \to u: \; L_{v,u}^{i,j}(t) = \frac{D_{i,j}(t)}{R_{v,u}(t)} + \kappa_{i,j}^{u}(t)\frac{C_{i,j}(t)D_{i,j}(t)}{f_{\text{cpu}}^{u}(t)}, \tag{5b}$$

$$u \to u': \; L_{u,u'}^{i,j}(t) = \frac{D_{i,j}(t)}{R_{u,u'}(t)} + \kappa_{i,j}^{u'}(t)\frac{C_{i,j}(t)D_{i,j}(t)}{f_{\text{cpu}}^{u'}(t)}, \tag{5c}$$

$$b \to u: \; L_{b,u}^{i,j}(t) = \frac{D_{i,j}(t)}{R_{b,u}(t)} + \kappa_{i,j}^{u}(t)\frac{C_{i,j}(t)D_{i,j}(t)}{f_{\text{cpu}}^{u}(t)}, \tag{5d}$$

$$u \rightarrow b: \; L_{u,b}^{i,j}(t) = \frac{D_{i,j}(t)}{R_{u,b}(t)} + \kappa_{i,j}^{b}(t)\frac{C_{i,j}(t)D_{i,j}(t)}{f_{\text{cpu}}^{b}(t)}, \quad (5e)$$

where the term $R_{x,y}(t)$ represents the data transfer rate between components x and y, where $\{x, y\} \in \{\mathcal{B} \cup \mathcal{V} \cup \mathcal{U}\}$ encompass all possible interactions within the BSs, vehicles, and UAVs. Additionally, $f_{\text{cpu}}^{z}(t)$, $z \in \{\mathcal{B} \cup \mathcal{U}\}$, signifies the CPU frequency of component x at time t. The binary variable $\kappa_{i,j}^{z}(t)$ with values in $\{0, 1\}$ indicates whether the task is being processed ($\kappa_{i,j}^{z}(t) = 1$) or not ($\kappa_{i,j}^{z}(t) = 0$) on component z. Specifically, in scenarios where $\kappa_{i,j}^{z}(t) = 0$, indicating the task is not processed at node z, the task is then forwarded to a connected node z'. This mechanism ensures dynamic task allocation and efficient utilization of processing capabilities across the network components.

As depicted in Fig. 1, vehicles in the network, represented as v, possess the capability to offload tasks to either a BS b or a UAV u. It is crucial to emphasize that each sub-task $\mathscr{T}_{i,j}$ is exclusively offloaded in entirety to a single entity, either to a BS or a UAV. Post offloading, there exists a provision for further forwarding the subtask $\mathscr{T}_{i,j}$ to an immediately connected node within the network. This means a BS might relay the task to an associated UAV, or conversely, a UAV might forward the task to a connected BS or another UAV, as illustrated in Fig. 1. However, such forwarding of tasks is carefully managed to align with the URLLC delay thresholds. There is a predefined limit to the extent of task forwarding to preclude any unwarranted delays. Within our multi-agent DRL framework, we have incorporated a sophisticated reward accumulation model. This model is designed to effectively mitigate excessive task relaying and to finely tune the task offloading process. The computation of the delay in this model, as referenced in (3), is described as:

$$L_{b}^{i,j}(t) = \sum_{u,b,u\neq u'} \left(\frac{D_{i,j}(t)}{R_{v,b}(t)} + \Gamma_{b,u}^{i,j}\frac{D_{i,j}(t)}{R_{b,u}(t)} + \Gamma_{u,u'}^{i,j}\frac{D_{i,j}(t)}{R_{u,u'}(t)} + \Gamma_{u,b}^{i,j}\frac{D_{i,j}(t)}{R_{u,b}(t)} + \frac{C_{i,j}(t)D_{i,j}(t)}{\max_z\left(f_{\text{cpu}}^{z}(t)\right)} \right), \forall\{i, j\}, \quad (6)$$

where $\Gamma_{b,u}^{i,j}$, $\Gamma_{u,u'}^{i,j}$, and $\Gamma_{u,b}^{i,j}$ are binary variables indicating the connectivity status between nodes. In a similar vein, if a vehicle's subtask is offloaded through a UAV, the corresponding latency $L_{u}^{i,j}(t)$ as per (3) is computed as:

$$L_{u}^{i,j}(t) = \sum_{u,b,u\neq u'} \left(\frac{D_{i,j}(t)}{R_{v,u}(t)} + \Gamma_{u,u'}^{i,j}\frac{D_{i,j}(t)}{R_{u,u'}(t)} + \Gamma_{u,b}^{i,j}\frac{D_{i,j}(t)}{R_{u,b}(t)} + \Gamma_{b,u}^{i,j}\frac{D_{i,j}(t)}{R_{b,u}(t)} + \frac{C_{i,j}(t)D_{i,j}(t)}{\max_z\left(f_{\text{cpu}}^{z}(t)\right)} \right), \forall\{i, j\}. \quad (7)$$

The DT estimates the available CPU frequencies, expressed as $\|f_{\text{cpu}}^{z}(t) - \overline{f_{\text{cpu}}^{z}}(t)\|$, where $\overline{f_{\text{cpu}}^{z}}(t)$ represents the estimated deviation (i.e., error) in CPU frequency for component z. These estimations are crucial for optimizing offloading strategies to minimize latency and enhance URLLC capabilities.

C. Mobility Model in VEC Network

The total duration of the timeframe $\widehat{T}$ is divided into T slots, each with duration τ such that $\widehat{T} = T\tau$. Each vehicle in $\mathcal{V}$ is allocated a sub-slot $t \in \mathcal{T}$ to request VEC services and initiate task offloading [4]. In our proposed model, vehicles and UAVs, excluding BSs, exhibit mobility governed by the random waypoint (RWP) model [19]. These entities are initially randomly distributed within a 3D space, bounded by $\mathcal{X} \in (-X_{\min}, +X_{\max})$, $\mathcal{Y} \in (-Y_{\min}, +Y_{\max})$, and $\mathcal{Z} \in (-Z_{\min}, +Z_{\max})$. We assume stationary positions for these elements during the time interval $\tau = [\Delta_{t,t-1}]$ [4]. In our modified RWP model, the vertical movement is influenced by both the speed and the angle of ascent or descent. This is particularly relevant for vehicles navigating hilly terrains or UAVs adjusting their altitude. The positions of vehicles and UAVs are updated at each time slot t as follows:

$$\tilde{x}_v(t) = \tilde{x}_v(t-1) + \tilde{V}_v(t-1)\Delta t \cos(\theta_v(t-1)), \quad (8)$$

$$\tilde{y}_v(t) = \tilde{y}_v(t-1) + \tilde{V}_v(t-1)\Delta t \sin(\theta_v(t-1)), \quad (9)$$

$$\tilde{z}_v(t) = \tilde{z}_v(t-1) + \tilde{V}_{v_z}(t-1)\Delta t \sin(\phi_v(t-1)), \quad (10)$$

where $\tilde{V}_v(t)$ and $\tilde{V}_{v_z}(t)$ represent the horizontal and vertical components of velocity, respectively. $\theta_v(t)$ is the horizontal movement direction, while $\phi_v(t)$ represents the angle of ascent or descent relative to the horizontal plane. For vehicles, this angle can represent the slope of the terrain, whereas for UAVs, it indicates the angle of altitude change. Hence, the location of the vehicle v at time instant t is denoted as $F_v(t) = \{\tilde{x}_v(t), \tilde{y}_v(t), \tilde{z}_v(t)\}, \forall v \in \mathcal{V}$. A similar approach is used for the UAVs, with their positions represented as $F_u(t) = \{\tilde{x}_u(t), \tilde{y}_u(t), \tilde{z}_u(t)\}, \forall u \in \mathcal{U}$. The location of each BS b is fixed and denoted as $F_b = \{\tilde{x}_b, \tilde{y}_b, \tilde{z}_b\}, \forall b \in \mathcal{B}$.

D. Data Rate Model and Transmission Delay

In our communication model, different transmission strategies are adopted based on the capabilities and roles of the involved entities as given in Table I.

TABLE I: Communication Models between Different Entities.

From	To	Mode	Channels	Data Rate
Vehicle v	BS b	SIMO	$\mathbf{h}_{v,b} \in \mathbb{C}^{M\times 1}$	$R_{v,b}(t)$
Vehicle v	UAV u	SISO	$h_{v,u} \in \mathbb{C}^{1\times 1}$	$R_{v,u}(t)$
UAV u	UAV u'	SISO	$h_{u,u'} \in \mathbb{C}^{1\times 1}$	$R_{u,u'}(t)$
BS b	UAV u	MISO	$\mathbf{h}_{b,u} \in \mathbb{C}^{1\times M}$	$R_{b,u}(t)$
UAV u	BS b	SIMO	$\mathbf{h}_{u,b} \in \mathbb{C}^{M\times 1}$	$R_{u,b}(t)$

1) Vehicle to BS Data Rate and Latency Calculation: Data transmission from a vehicle v to a BS b (each BS has M antennas) in a vehicular network is influenced by path-loss, Doppler effects due to mobility, imperfect CSI, and the angle-of-arrival (AoA). Channel gain $\mathbf{h}_{v,b}(t) \in \mathbb{C}^{M\times 1}$ is written as:

$$\mathbf{h}_{v,b}(t) = \sqrt{\text{PL}_{v,b}(t)}e^{j2\pi f_{d,v}(t)\tau}(\widehat{\mathbf{h}}_{v,b}(t) + \epsilon_{v,b}(t)), \quad (11)$$

where $\text{PL}_{v,b}(t)$ represents the path-loss. To incorporate this parameter accurately in our calculations, we need to transform it from a logarithmic (dB) to a linear scale. This conversion is achieved by applying the formula $10^{\frac{\text{PL}_{v,u}(t)}{10}}$, where $\text{PL}_{v,u}(t)$ is the path-loss in dB. $f_{d,v}(t)$ is the Doppler shift, $\widehat{\mathbf{h}}_{v,b}(t)$ is the estimated CSI. The term $\epsilon_{v,b}(t) \in \mathbb{C}^{M\times 1}$ in (11) is the CSI estimation error. It is characterized by the set as follows [20]:

$$\mathcal{T}_{\epsilon} = \left\{\epsilon_{v,b} \in \mathbb{C}^{M\times 1} : \|\epsilon_{v,b}\| \leq \tau_{v,b},\ v, b = 1, \ldots, M\right\}, \quad (12)$$

where $\|\cdot\|$ denotes the norm. Within the set, entries of $\epsilon_{v,b}$ are independent and identically distributed (i.i.d) and assumed to have zero mean with variance $\sigma^2_{v,b}$. For the present work, we consider uniformly distributed bounded CSI uncertainty $\sigma^2_{v,b} = \{0.05, 0.10\}$, as discussed in [20], [21].

The Rayleigh fading scenario between the vehicle v and the BS b implies the absence of the LoS path. Thus, the channel $\widehat{\mathbf{h}}_{v,b}(t)$ is written as:

$$\widehat{\mathbf{h}}_{v,b}(t) = \mathbf{a}_v\left(\vartheta_v(t)\right), \tag{13}$$

where $\mathbf{a}_v\left(\vartheta_v(t)\right)$ represents the steering vector at the BS's multiple antennas M and expressed as [22]:

$$\mathbf{a}_v\left(\vartheta_v(t)\right) = \left[1, e^{j\frac{2\pi\delta}{\lambda(t)}\sin\vartheta_v(t)}, .., e^{j\frac{2\pi\delta}{\lambda(t)}(M-1)\sin\vartheta_v(t)}\right]^{\mathrm{T}}, \tag{14}$$

with δ as the antenna separation distance and $\vartheta_v(t)$ as the AoA. This specification highlights that our focus is on the geometric aspects of the signal reception, particularly how the signal's arrival angle at the BS antennas affects the observed signal [22], without directly addressing the stochastic nature of the channel gain's magnitude in the above equation.

The urban path-loss (in dB) is given by using the Okumura model [23]:

$$\begin{aligned}\mathrm{PL}_{v,u}(t) =& 69.55 + 26.16\log_{10} f_c - 13.82\log_{10}(h_{\mathrm{BS}}) - a(h_{\mathrm{veh}}) \\ &+ (44.9 - 6.55\log_{10}(h_{\mathrm{BS}}))\log_{10}(d_{v,b}(t)),\end{aligned} \tag{15}$$

where f_c is the carrier frequency in MHz, h_{BS} is the height of the BS's antenna above the ground, and $a(h_{\mathrm{veh}})$ is a correction factor for the height of the vehicle's antenna, calculated as $a(h_{\mathrm{veh}}) = (1.1\log_{10}(f_c) - 0.7)h_{\mathrm{veh}} - (1.56\log_{10}(f_c) - 0.8)$. The symbol $d_{v,b}(t)$ represents the distance between the vehicle v and the BS b at time t.

However, in the context of vehicle-to-BS communications, the 3GPP technical specifications provide detailed path-loss models catering to various environmental conditions [24]. Among these, the urban macro (UMa) path-loss models are pivotal for evaluating vehicle-to-infrastructure communications. These models are differentiated based on LoS and NLoS propagation conditions, which are expressed as [24]:

$$\mathrm{PL}_{\mathrm{LoS}} = \begin{cases}\mathrm{PL}_1, & \text{for } 10\,\mathrm{m} \le d_{v,b} \le d'_{\mathrm{BP}}, \\ \mathrm{PL}_2, & \text{for } d'_{\mathrm{BP}} < d_{v,b} \le 5\,\mathrm{km},\end{cases} \tag{16}$$

$$\mathrm{PL}_1 = 28.0 + 22\log_{10}(d_{v,b}) + 20\log_{10}(f_c), \tag{17}$$

$$\begin{aligned}\mathrm{PL}_2 =& 28.0 + 40\log_{10}(d_{v,b}) + 20\log_{10}(f_c) \\ &- 9\log_{10}\left((d'_{\mathrm{BP}})^2 + (h_{\mathrm{BS}} - h_{\mathrm{UT}})^2\right),\end{aligned} \tag{18}$$

$$\mathrm{PL}_{\mathrm{NLoS}} = \max\left(\mathrm{PL}_{\mathrm{LoS}}, \mathrm{PL}'_{\mathrm{NLoS}}\right), \tag{19}$$

$$\begin{aligned}\mathrm{PL}'_{\mathrm{NLoS}} =& 13.54 + 39.08\log_{10}(d_{v,b}) + 20\log_{10}(f_c) \\ &- 0.6(h_{\mathrm{UT}} - 1.5),\end{aligned} \tag{20}$$

where d'_{BP} is the breakpoint distance for LoS. h_{UT} is the height of the vehicle's antenna.

The Doppler shift due to the relative velocity between vehicle v and the stationary BS b is computed as:

$$f_{d,v}(t) = \frac{\widehat{V}_v(t)\cos(\varphi_v(t))}{\lambda}, \tag{21}$$

where $\widehat{V}_v(t)$ is the relative velocity, $\varphi_v(t) \sim \mathcal{U}(-90°, +90°)$ is the angle of vehicle movement relative to the LoS, and λ is the wavelength of the carrier signal. The signal-to-interference-plus-noise ratio (SINR) at the BS for the signal received from vehicle v is given by:

$$\omega_{v,b}(t) = \frac{p_{v,b}(t)\left|\mathbf{h}_{v,b}(t)\right|^2}{\sigma_b^2}, \tag{22}$$

where $p_{v,b}(t)$ is the transmit power from vehicle v to BS b and σ_b^2 is the noise power at BS.

The URLLC achievable rate $R_{v,b}(t)$ for communication from vehicle v to BS b is calculated as:

$$R_{v,b}(t) = \mathscr{B}\left(\log_2\left(1 + \omega_{v,b}(t)\right) - \sqrt{\frac{V(\omega_{v,b}(t))}{l_v(t)}} \times \zeta\right), \tag{23}$$

where $\mathscr{B}$ is the system bandwidth, $V(\omega_{v,b}(t))$ represents the channel dispersion, $l_v(t)$ is the finite block length, and $\zeta = \frac{\mathrm{Q}^{-1}(\epsilon_b)}{\log_e 2}$, where $\mathrm{Q}^{-1}(\epsilon_b)$ is the inverse of the Gaussian Q-function with parameter ϵ_b used to calculate the decoding error probability. Finally, the transmission latency for the sub-task $\mathscr{T}_{i,j}$ from vehicle v to BS b is determined as: $\mathscr{T}_v^b(t) = \frac{D_{i,j}(t)}{R_{v,b}(t)}$. It is worth mentioning here that the convergence of transmission time in SISO over Rayleigh fading channels is not directly applicable to the scope and methodologies employed in our research [25]. Our analysis implicitly assumes an MISO setting that mitigates the severe effects of deep fades inherent to Rayleigh fading [25]. This approach allows us to maintain the assumption of a finite average transmission time, which is crucial for the practical significance of our latency calculations. Specifically, for the latency calculation from the vehicle to the base station in (23), we consider $M > 1$ for the MISO model, which ensures a finite expected transmission time in our analyses [25].

2) Vehicle to UAV Data Rate and Latency Calculation: We adopt a Rician channel model for the vehicle-to-UAV communication, considering the combination of LoS and non-line-of-sight (NLoS) components due to the potential for a clear LoS. The channel gain $h_{v,u}(t)$ between vehicle v and UAV u is modeled as:

$$\begin{aligned}h_{v,u}(t) = \sqrt{\mathrm{PL}_{v,u}(t)}\Bigg(&\sqrt{\frac{K}{K+1}}h_{\mathrm{LoS}}(t)e^{j2\pi f_{d,v,u}(t)\tau} \\ &+ \sqrt{\frac{1}{K+1}}h_{\mathrm{NLoS}}(t)\Bigg),\end{aligned} \tag{24}$$

where K is the Rician factor, and we set $K = 6$ [26], [27]. The Doppler shift $f_{d,u,u'}(t)$ is modeled as given in (21). The LoS component $h_{\mathrm{LoS}}(t)$ and the NLoS component $h_{\mathrm{NLoS}}(t)$ are given respectively by:

$$h_{\mathrm{LoS}}(t) = e^{j\phi(t)}, \quad h_{\mathrm{NLoS}}(t) = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} e^{j\theta_i(t)}, \tag{25}$$

where $\phi(t)$ and $\theta_i(t)$ represent the random phases for the LoS and the i-th NLoS paths, respectively, and N is the number of NLoS paths. The free-space path-loss is given by:

$$\mathrm{PL}_{v,u}(t) = \mathrm{PL}_0\left(\frac{d_{v,u}(t)}{d_0}\right)^{-\alpha_{PL}}, \tag{26}$$

where $d_{v,u}(t)$ is the distance between the vehicle and the UAV, and α_{PL} is the path-loss exponent.

An advanced 3GPP path-loss model [28] in an UMa scenario for UAV communication measures the signal attenuation in UAV-to-ground and UAV-to-UAV communication scenarios, where the altitude of UAVs and the environment play significant roles in determining the quality of the communication link. These are defined as follows [28]:

$$\mathrm{PL}_{\mathrm{LoS}} = 28 + 22\log_{10}(d) + 20\log_{10}(f_c), \tag{27}$$

$$\mathrm{PL}_{\mathrm{NLoS}} = -17.5 + (46 - 7\log_{10}(h_R))\log_{10}(d_{v,u}) + 20\log_{10}(40\pi f_c/3), \tag{28}$$

$$\mathcal{P}_{\mathrm{LoS}} = \begin{cases} 1, & d_{v,u} \le d_1, \\ \frac{d_1}{d_{v,u}} + \left(1 - \frac{d_1}{d_{v,u}}\right)\exp\left(-\frac{d_{v,u}}{p_1}\right), & d_{v,u} > d_1, \end{cases} \tag{29}$$

where $\mathcal{P}_{\mathrm{LoS}}$ is the probability of having a LoS condition between the transmitter and receiver. h_R is the height of the UAV. d_1 is the threshold distance from the transmitter to the receiver. A smaller p_1 value indicates a faster decrease in LoS probability, suggesting that LoS conditions become less likely at shorter distances beyond d_1. According to [24], we estimate to consider $p_1 = 63$ for the present work. The above LoS probability is calculated by considering the fact that the UAV's height belongs to a range of 22.5–100 meters. Once a UAV is flying higher than $d_1 = 100$ meters in height, the channel becomes LoS [28].

The SINR for the vehicle-to-UAV link is calculated as follows:

$$\omega_{v,u}(t) = \frac{p_{v,u}(t)|\hat{h}_{v,u}(t)|^2}{\sigma_u^2}, \tag{30}$$

where $p_{v,u}(t)$ is the transmit power from vehicle v to UAV u and σ_u^2 is the noise power. The channels follow imperfect CSI as discussed in (13).

The URLLC achievable data rate $R_{v,u}(t)$ for communication from vehicle v to UAV u is:

$$R_{v,u}(t) = \mathscr{B}\left(\log_2(1+\omega_{v,u}(t)) - \sqrt{\frac{V(\omega_{v,u}(t))}{l_v(t)}} \times \zeta\right). \tag{31}$$

The transmission latency $\mathscr{T}_v^u(t)$ for a data packet of size $D_{i,j}(t)$ from vehicle v to UAV u is given by: $\mathscr{T}_v^u(t) = \frac{D_{i,j}(t)}{R_{v,u}(t)}$.

3) UAV to UAV Data Rate and Latency Calculation: For UAV-to-UAV communication, we use a Rician channel model, which is affected by the Doppler shift due to the relative motion of UAVs. The channel gain $h_{u,u'}(t)$ between UAV u and UAV u' is modeled as (24) and the path-loss is modeled similar to (26). The SINR for the UAV-to-UAV link is:

$$\omega_{u,u'}(t) = \frac{P_{u,u'}(t)|h_{u,u'}(t)|^2}{\sigma_{u'}^2}, \tag{32}$$

where $P_{u,u'}(t)$ is the transmit power and $\sigma_{u'}^2$ is the noise power. The achievable data rate $R_{u,u'}(t)$ is calculated as:

$$R_{u,u'}(t) = \mathscr{B}\left(\log_2(1+\omega_{u,u'}(t)) - \sqrt{\frac{V(\omega_{u,u'}(t))}{l_v(t)}} \times \zeta\right). \tag{33}$$

The transmission latency $\mathscr{T}_u^{u'}(t)$ for a data packet of size $D_{i,j}(t)$ from UAV u to UAV u' is: $\mathscr{T}_u^{u'}(t) = \frac{D_{i,j}(t)}{R_{u,u'}(t)}$.

4) BS-to-UAV and UAV-to-BS Rate and Latency Model: In our vehicular network model, BS-to-UAV and UAV-to-BS communications follow the Rician channel model to accommodate the mixed LoS and NLoS scenarios. The Doppler effect due to relative mobility is also considered, particularly significant in dynamic UAV environments.

a) BS-to-UAV (MISO): The BS b communicates with UAV u using a MISO model. The channel gain between BS and UAV at time instant t is calculated similarly to the (24). The SINR at the UAV is calculated as:

$$\omega_{b,u}(t) = \frac{P_{b,u}(t)|\mathbf{h}_{b,u}(t)|^2}{\sigma_u^2}, \tag{34}$$

where $\mathbf{h}_{b,u} \in \mathbb{C}^{M\times 1}$, $P_{b,u}(t)$ is the transmit power and σ_u^2 is the noise power.

The URLLC achievable rate $R_{b,u}(t)$ is given by:

$$R_{b,u}(t) = \mathscr{B}\left(\log_2(1+\omega_{b,u}(t)) - \sqrt{\frac{V(\omega_{b,u}(t))}{l_b(t)}} \times \zeta\right), \tag{35}$$

and the latency $\mathscr{T}_b^u(t)$ is: $\mathscr{T}_b^u(t) = \frac{D_{i,j}(t)}{R_{b,u}(t)}$.

b) UAV-to-BS (SIMO): We use a SIMO model for UAV u communicating with BS b. The channel gain $h_{u,b}(t)$ is modeled as similar to (24). The SINR at the BS is written as

$$\omega_{u,b}(t) = \frac{P_{u,b}(t)|\mathbf{h}_{u,b}(t)|^2}{\sigma_b^2}, \tag{36}$$

where $\mathbf{h}_{u,b}(t) \in \mathbb{C}^{1\times M}$. The URLLC data rate $R_{u,b}(t)$ is:

$$R_{u,b}(t) = \mathscr{B}\left(\log_2(1+\omega_{u,b}(t)) - \sqrt{\frac{V(\omega_{u,b}(t))}{l_u(t)}} \times \zeta\right), \tag{37}$$

and the transmission latency $\mathscr{T}_u^b(t)$ is: $\mathscr{T}_u^b(t) = \frac{D_{i,j}(t)}{R_{u,b}(t)}$.

III. Problem Formulation

The objective of the problem is to minimize the total task offloading latency in the URLLC-enabled VEC network. In comparison to the task offloading process for MEC, it is observed that the results are returned in a significantly smaller number of bits. Consistent with the findings presented in [14], the time taken to transmit the computational outcomes back to the vehicle is negligible. As a result, and in line with the observations in [14], [29], we can reasonably disregard the time and power required for transmitting these computation results back to the vehicle, given the substantially smaller size of the computation outcomes compared to the volume of data offloaded. The objective of the problem is to optimize the overall performance of the URLLC-enabled VEC network as mentioned in (3). However, considering multiple factors such as latency, energy efficiency, and quality-of-service (QoS) ensures a more efficient and reliable network operation. The objective function is thus formulated to minimize not only the total latency for task offloading but also to incorporate energy consumption and QoS considerations:

$$\min_{\mathbf{a}} \sum_{v\in\mathcal{V}}\sum_{i,j} \mathscr{P}_{i,j}(t)\left(\sum_{b\in\mathcal{B}} \Gamma_{v,b}^{i,j}(t) L_b^{i,j}(t) + \sum_{u\in\mathcal{U}} \Gamma_{v,u}^{i,j}(t) L_u^{i,j}(t)\right.$$

$$+\lambda_E \sum_{v\in\mathcal{V}}\sum_{i,j} E_v^{i,j}(t) - \lambda_Q \sum_{v\in\mathcal{V}}\sum_{i,j} Q_{i,j}(t)\Bigg), \quad (38)$$

where $\mathbf{a}$ is a collective action space for the network and is written as $\mathbf{a} = \{\mathbf{a}_v \cup \mathbf{a}_u \cup \mathbf{a}_b\}$, with $\mathbf{a}_v$, $\mathbf{a}_u$, and $\mathbf{a}_b$ as the action spaces of the vehicle, UAV, and BS agents, respectively. A detailed exposition of each agent's action space, along with the parameters and functionalities encompassed within, is provided in Section IV. λ_E and λ_Q are the weighting factors that balance the importance of energy consumption and QoS against latency in the objective function, $\lambda_E + \lambda_Q = 1$. The energy consumption $E_v^{i,j}(t)$ associated with offloading subtask $\mathscr{T}_{i,j}$ from vehicle v at time t, and $Q_{i,j}(t)$ denotes the QoS for the same task. Hereafter, we use the following constraints to maintain the operational integrity and efficiency of the system, ensuring that all processes align with the predefined performance, resource allocation, and energy consumption standards:

1) Connectivity Constraints: Each vehicle must be connected to at least one BS or UAV:

$$C1: \forall v \in \mathcal{V}, \exists b \in \mathcal{B}\cup\mathcal{U} : \Gamma_{v,b}^{i,j} = 1 \quad \text{or} \quad \Gamma_{v,u}^{i,j} = 1. \quad (39)$$

2) Task Offloading Constraints: Each subtask $\mathscr{T}_{i,j}$ of a vehicle can be offloaded to either a BS or a UAV, but not both:

$$C2: \forall v \in \mathcal{V}, \forall i,j : \Gamma_{v,b}^{i,j} + \Gamma_{v,u}^{i,j} \le 1. \quad (40)$$

3) Resource Constraints: The computational demand of offloaded task $\mathscr{T}_{i,j}$ should not exceed the processing capacity ($\mathscr{C}$) of BSs or UAVs.

$$C3: \sum_{v\in\mathcal{V}}\sum_{i,j} C_{i,j}(t) D_{i,j}(t) \Gamma_{v,b}^{i,j} \le \mathscr{C}_b, \quad \forall b \in \mathcal{B}. \quad (41)$$

$$C4: \sum_{v\in\mathcal{V}}\sum_{i,j} C_{i,j}(t) D_{i,j}(t) \Gamma_{v,u}^{i,j} \le \mathscr{C}_u, \quad \forall u \in \mathcal{U}. \quad (42)$$

4) Latency Constraints: The latency for the subtask $\mathscr{T}_{i,j}$ must not exceed a specified threshold ϱ:

$$C5: \forall\{i,j\}, \forall v \in \mathcal{V} : L_b^{i,j}(t) \le \varrho, \quad \text{if} \quad \Gamma_{v,b}^{i,j} = 1. \quad (43)$$

$$C6: \forall\{i,j\}, \forall v \in \mathcal{V} : L_u^{i,j}(t) \le \varrho, \quad \text{if} \quad \Gamma_{v,u}^{i,j} = 1. \quad (44)$$

5) CPU Frequency Constraints: The CPU frequency of BSs and UAVs must be within the permissible range:

$$C7: f_{\min} \le f_{\text{cpu}}^{z}(t) \le f_{\max}^{z}, \quad \forall z \in \mathcal{B}\cup\mathcal{U}. \quad (45)$$

6) Data Rate Constraints: The data rate for each link must satisfy the channel capacity $\widehat{R}(t)$:

$$C8: \widehat{R}(t) \le R_{x,y}(t), \quad \forall\{x,y\} \in \{\mathcal{B}\cup\mathcal{V}\cup\mathcal{U}\}. \quad (46)$$

7) Energy Efficiency Constraints: Minimizing the energy consumption for task offloading. The energy consumption $E_v^{i,j}(t)$ for offloading subtask $\mathscr{T}_{i,j}$ from vehicle v at time t is given by: $E_v^{i,j}(t) = \frac{D_{i,j}(t)}{P_v^{i,j}(t)}, \quad \forall v \in \mathcal{V}, \forall i,j$, where $P_v^{i,j}(t)$ is the power consumption and $\mathscr{T}_v^{i,j}(t)$ is the time taken for offloading subtask $\mathscr{T}_{i,j}$. The total energy consumption for all vehicles in the network should not exceed a predefined maximum energy budget $E_{\max}$:

$$C9: \sum_{i,j} E_v^{i,j}(t) \le E_{\max}^{v}, \quad \forall v \in \mathcal{V}. \quad (47)$$

8) QoS Constraints: Ensuring the QoS for each task offloaded to the network involves considering latency, data rate, and reliability. We define a QoS function as:

$$C10: Q_{i,j}(t) = \frac{w_L}{L_{i,j}(t)} - w_E E_v^{i,j}(t) \ge Q_{\min}, \quad \forall i,j, \quad (48)$$

where $L_{i,j}(t)$ is the latency of subtask $\mathscr{T}_{i,j}$ at time t, $R_{i,j}(t)$ is the data rate for subtask $\mathscr{T}_{i,j}$ at time t, and w_L and w_E are the weights assigned to latency and data rate, respectively. The weights are chosen to balance the impact of each factor on the overall QoS. The inverse of latency ensures that lower latency contributes positively to the QoS. The QoS is considered adequate if it exceeds a minimum threshold $Q_{\min}$.

9) Network Stability Constraints: Ensuring the stability of the network involves controlling the rate of change in offloading decisions, as well as maintaining a balanced load distribution among the network nodes. The stability is defined by the following constraints:

a) Offloading Decision Variability: The offloading decision variability is managed using probabilistic functions $\mathcal{P}_{v,b}^{i,j}(t)$ and $\mathcal{P}_{v,u}^{i,j}(t)$, representing the probability of change in offloading decisions to a BS b or a UAV u for vehicle v: $\mathcal{P}_{x,y}^{i,j}(t) = 1/\left(1 + e^{-k(|\Gamma_{x,y}^{i,j}(t) - \Gamma_{x,y}^{i,j}(t-1)| - \Delta S_{\max})}\right)$. The constraints are then defined as:

$$C13: \mathcal{P}_{v,b}^{i,j}(t) \le \Delta S_{\max}, \quad \forall v \in \mathcal{V}, \forall b \in \mathcal{B}. \quad (49)$$

$$C14: \mathcal{P}_{v,u}^{i,j}(t) \le \Delta S_{\max}, \quad \forall v \in \mathcal{V}, \forall u \in \mathcal{U}. \quad (50)$$

b) Load Distribution Constraints: To prevent overloading any single node (BS or UAV) in the network, the load distribution is controlled as follows:

$$C14: \sum_{v\in\mathcal{V}}\sum_{i,j} \Gamma_{v,b}^{i,j}(t) L_b^{i,j}(t) \le \varrho, \quad \forall b \in \mathcal{B}, \quad (51)$$

$$C15: \sum_{v\in\mathcal{V}}\sum_{i,j} \Gamma_{v,u}^{i,j}(t) L_u^{i,j}(t) \le \varrho, \quad \forall u \in \mathcal{U}, \quad (52)$$

where $L_b^{i,j}(t)$ and $L_u^{i,j}(t)$ are the latencies for processing subtask $\mathscr{T}_{i,j}$ through BS b and UAV u respectively. These constraints ensure that the network can adapt to changes in offloading decisions while maintaining a balanced load and preventing rapid fluctuations that could destabilize the network operations.

10) Power Constraints for Intermediate Transmission: To ensure sustainable operation and energy efficiency in the network, we impose power constraints for both transmission and processing at BSs and UAVs. The transmission power of each BS and UAV should not exceed their respective maximum power limits:

$$C16: P_b(t) \le P_{\max,b}, \; C17: \; P_u(t) \le P_{\max,u}, \; \forall b,u, \quad (53)$$

where $P_b(t)$ and $P_u(t)$ are the transmission powers of BS b and UAV u at time t, respectively. $P_{\max,b}$ and $P_{\max,u}$ are the maximum allowable transmission powers for the respective BS and UAV.

11) Energy Constraints for Task Processing: The energy consumption for processing tasks at the BS and UAV must be managed efficiently to adhere to a maximum energy budget. The energy required for processing a task of size $D_{i,j}$ with complexity $C_{i,j}$ at the BS and UAV is governed by the following constraints:

$$C18: \sum_{i,j} E^b_{i,j}(t) = \sum_{i,j} \xi_b (CD)_{i,j}(t) \leq E^b_{\max}, \forall b \in \mathcal{B}, \quad (54)$$

$$C19: \sum_{i,j} E^u_{i,j}(t) = \sum_{i,j} \xi_u (CD)_{i,j}(t) \leq E^u_{\max}, \forall u \in \mathcal{U}, \quad (55)$$

where $E^b_{\max}$ and $E^u_{\max}$ represent the maximum allowable energy budget for task processing at each BS and UAV, respectively. ξ_b and ξ_u are the energy consumption coefficients per unit data and complexity for the BS and UAV, respectively.

We incorporate an additional constraint to ensure efficient task load distribution among all subtasks. This constraint regulates the size of each subtask, maintaining an upper limit to optimize the distribution process. Mathematically, the constraint is formulated as:

$$C20: D_{i,j} \leq D^{\max}_{i,j}, \quad (56)$$

where $D^{\max}_{i,j}$ represents the maximum data size of a subtask.

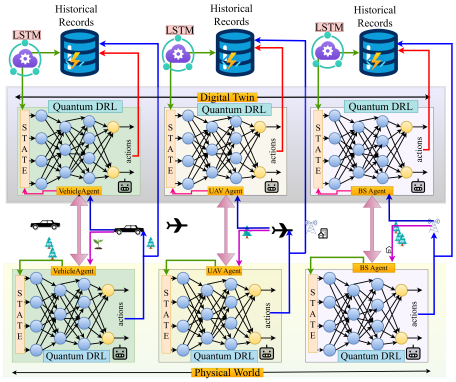

Fig. 2: Architecture of DT-aided Quantum-DRL-based solution.

IV. MDP Formulation in Multi-Agent Framework

The integration of Quantum-DRL within a DT framework, assisted by LSTM networks, forms an effective solution for optimizing the present task offloading in URLLC-enabled VEC networks. To deploy this strategy, the optimization problem is modeled as a Markov decision process (MDP), where the state space $\mathcal{S}$ represents network conditions and actions $\mathcal{A}$ denote task offloading and resource allocation decisions. Furthermore, a transition model $T(s, a, s')$ captures the probabilistic transition of the network's state. The Quantum-DRL utilizes a quantum-enhanced policy $\pi(a|s)$ to select actions that maximize the expected reward $\mathbb{E}[\mathcal{R}(s, a)]$, while LSTM networks in DT provide a temporal context by analyzing historical patterns. Our proposed solution model, as depicted in Fig. 2, adopts a multi-agent approach with vehicles ($\mathcal{V}$), UAVs ($\mathcal{U}$), and BSs ($\mathcal{B}$) as agents in an urban URLLC-enabled VEC network. Each agent interacts within this network to optimize task offloading decisions under strict latency constraints, considering a Nash equilibrium framework within a Markovian game model. To apply Quantum DRL, we define state spaces and action spaces for vehicles, UAVs, and BSs agents as follows:

1) Vehicle Agent ($\mathscr{A}_{veh}$): For vehicles ($v \in \mathcal{V}$), the state and action spaces are comprised of stochastic variables pertinent to the vehicle's operation within the network. These spaces are defined as follows:

- **State Space ($\mathcal{S}_v$):** The state of vehicle v at time t, denoted as $\mathbf{s}_v(t) \in \mathcal{S}_v$, is characterized by a tuple of variables including its location $F_v(t)$, velocity attributes such as $\tilde{V}_v(t), \tilde{V}_{v_z}(t), \theta_v(t), \phi_v(t)$, CSI $\mathbf{h}_{v,b}(t)$ and $h_{v,u}(t)$, as well as the path-loss indicators $\mathrm{PL}_{v,b}(t)$ and $\mathrm{PL}_{v,u}(t)$. So, the state space for vehicle v at time t, denoted as $\mathbf{s}_v(t)$, is defined as:

$$\mathbf{s}_v(t) = \{F_v(t), \tilde{V}_v(t), \tilde{V}_{v_z}(t), \theta_v(t), \phi_v(t), \mathbf{h}_{v,b}(t), h_{v,u}(t), \mathrm{PL}_{v,b}(t), \mathrm{PL}_{v,u}(t)\}. \quad (57)$$

- **Action Space ($\mathcal{A}_v$):** The action space $\mathbf{a}_v(t) \in \mathcal{A}_v$ for vehicle v at time t comprises a set of control parameters that directly or indirectly affect the decision-making process of the agent $\mathscr{A}_{\text{veh}}$. The capabilities of $\mathscr{A}_{\text{veh}}$ include managing the transmission power level $P^{i,j}_v(t)$, making decisions on task processing designated by $\Gamma^{i,j}_{v,b}(t) = \{0, 1\}$ for backend tasks and $\Gamma^{i,j}_{v,u}(t) = \{0, 1\}$ for user-related tasks. Additionally, $\mathscr{A}_{\text{veh}}$ is responsible for task prioritization, indicated by $\mathscr{P}_{i,j}(t)$, and determining the data segment size $D_{i,j}$. These actions are selected following network constraints and operational objectives. Therefore, the action space for $\mathscr{A}_{\text{veh}}$ at any given time instant t is represented as:

$$\mathbf{a}_v(t) = \{P^{i,j}_v(t), \Gamma^{i,j}_{v,b}(t), \Gamma^{i,j}_{v,u}(t), \mathscr{P}_{i,j}(t), D_{i,j}(t)\}. \quad (58)$$

2) UAV Agent ($\mathscr{U}_{uav}$): The UAV agent $\mathscr{U}_{\text{uav}}$ plays a pivotal role in the decision-making framework of the network, necessitating a well-structured state and action space to define its capabilities and responsibilities.

- **State Space ($\mathcal{S}_u$):** The state space $\mathbf{s}_u(t) \in \mathcal{S}_u$ of the UAV agent $\mathscr{U}_{\text{uav}}$ incorporates a detailed set of parameters reflecting the UAV's status and environmental interactions. It includes the UAV's spatial coordinates $F_u(t)$, horizontal and vertical velocities $\tilde{U}_u(t)$ and $\tilde{U}_{u_z}(t)$, and angular orientations $\theta_u(t)$ and $\phi_u(t)$. Communication factors are represented by CSI $\mathbf{h}_{u,b}(t)$ between the UAV and BSs and $h_{u,u'}(t)$ for UAV-to-UAV links. Path-loss indicators $\mathrm{PL}_{u,b}(t)$ and $\mathrm{PL}_{u,u'}(t)$ denote transmission efficiencies. The state space also encompasses task data size $D_{i,j}(t)$, task complexity $C_{i,j}(t)$, and available CPU frequency $f^u_{\max}(t)$, essential for computational decision-making.

$$\mathbf{s}_u(t) = \{F_u(t), \tilde{U}_u(t), \tilde{U}_{u_z}(t), \theta_u(t), \phi_u(t), \mathbf{h}_{u,b}(t), h_{u,u'}(t), \mathrm{PL}_{u,b}(t), \mathrm{PL}_{u,u'}(t), D_{i,j}(t), C_{i,j}(t), f^z_{\max}(t)\}. \quad (59)$$

- **Action Space ($\mathcal{A}_u$):** The action space $\mathbf{a}_u(t) \in \mathcal{A}_u$ of $\mathscr{U}_{\text{uav}}$ delineates the UAV's potential actions in response to its observed state. These actions include managing the power allocation $P_u(t)$ for task forwarding to either BS

or other UAVs, selecting the CPU frequency $f^u_{\text{cpu}}(t)$ for task processing at the MEC, making decisions on task processing $\kappa^u_{i,j} = \{0, 1\}$, and determining task forwarding paths via binary variables $\Gamma^{i,j}_{u,u'}$ and $\Gamma^{i,j}_{u,b}$. Additionally, considering the UAV's role in the network, actions related to maintaining communication quality, energy efficiency, and adherence to operational constraints are also integral to $\mathcal{A}_u$. Therefore, the action vector $\mathbf{a}_u(t)$ is written as:

$$\mathbf{a}_u(t) = \{P_u(t), f^u_{\text{cpu}}(t), \kappa^u_{i,j}, \Gamma^{i,j}_{u,u'}, \Gamma^{i,j}_{u,b}\}. \quad (60)$$

3) BS Agent ($\mathscr{B}_{bs}$): The BS agent, denoted as $\mathscr{B}_{\text{bs}}$, is an integral component of the URLLC-enabled VEC network. It is responsible for managing task offloading from vehicles, coordinating with UAVs, and ensuring efficient utilization of network resources. The state and action spaces for the BS agent are defined as follows:

- **State Space ($\mathcal{S}_b$):** The state space $\mathbf{s}_b(t) \in \mathcal{S}_b$ for a BS agent at time t includes parameters that reflect the BS's operational status and its interactions with vehicles and UAVs. This encompasses the BS's location F_b, the CSI with vehicles $\mathbf{h}_{v,b}(t)$, and with UAVs $\mathbf{h}_{u,b}(t)$. Additionally, path-loss indicators $\text{PL}_{v,b}(t)$ and $\text{PL}_{u,b}(t)$, representing the transmission efficiency to vehicles and UAVs respectively, are included. The state space also considers the task data size $D_{i,j}(t)$, task complexity $C_{i,j}(t)$, and the available CPU frequency $f^b_{\max}(t)$, which are crucial for task processing decision.

$$\mathbf{s}_b(t) = \{F_b, \mathbf{h}_{v,b}(t), \mathbf{h}_{u,b}(t), \text{PL}_{v,b}(t), \text{PL}_{u,b}(t), D_{i,j}(t), C_{i,j}(t), f^b_{\max}(t)\}. \quad (61)$$

- **Action Space ($\mathcal{A}_b$):** The action space $\mathbf{a}_b(t) \in \mathcal{A}_b$ of $\mathscr{B}_{\text{bs}}$ includes adjusting the transmission power $P_b(t)$, deciding on task processing and offloading $\kappa^b_{i,j} = \{0, 1\}$, managing data forwarding $\Gamma^{i,j}_{b,u}$ to UAVs, and optimizing the CPU frequency allocation $f^b_{\text{cpu}}(t)$ for task processing. This set of actions is essential to ensure efficient task processing and is written as

$$\mathbf{a}_b(t) = \{P_b(t), \kappa^b_{i,j}, \Gamma^{i,j}_{b,u}, f^b_{\text{cpu}}(t)\}. \quad (62)$$

A. Multi-Agent Associative Reward Function Formulation

Given the multi-agent setup in our URLLC-enabled VEC network, we define an associative reward function $\mathscr{R}(\mathbf{s}, \mathbf{a})$ to encourage actions that minimize latency, optimize energy consumption, and enhance the QoS. This function is designed to reflect the Nash equilibrium strategy [30], ensuring that no agent benefits by unilaterally changing its strategy. The overall reward function is a composition of individual rewards for vehicles ($\mathscr{R}_{\mathcal{V}}$), UAVs ($\mathscr{R}_{\mathcal{U}}$), and BSs ($\mathscr{R}_{\mathcal{B}}$):

$$\mathscr{R}(\mathbf{s}, \mathbf{a}) = \mathscr{R}_{\mathcal{V}}(\mathbf{s}_v, \mathbf{a}_v, \mathbf{s}_{-v}, \mathbf{a}_{-v}) + \mathscr{R}_{\mathcal{U}}(\mathbf{s}_u, \mathbf{a}_u, \mathbf{s}_{-u}, \mathbf{a}_{-u}) + \mathscr{R}_{\mathcal{B}}(\mathbf{s}_b, \mathbf{a}_b, \mathbf{s}_{-b}, \mathbf{a}_{-b}), \quad (63)$$

where $\mathscr{R}(\mathbf{s}, \mathbf{a})$ represents the total reward function. The individual reward functions for vehicles, UAVs, and BSs are denoted by $\mathscr{R}_{\mathcal{V}}$, $\mathscr{R}_{\mathcal{U}}$, and $\mathscr{R}_{\mathcal{B}}$, respectively. In these expressions, $\mathbf{s}_x$ and $\mathbf{a}_x$ represent the states and actions of agents of type x, while $\mathbf{s}_{-x}$ and $\mathbf{a}_{-x}$ represent the states and actions of all other agents not of type x, respectively.

1) Vehicle Reward Function ($\mathscr{R}_{\mathcal{V}}$): The reward function for vehicle agents includes the optimization of latency, energy efficiency, and QoS. This function integrates these aspects with tailored weights, enabling a comprehensive evaluation of vehicle agents' performance. The formulation is as follows:

$$\mathscr{R}_{\mathcal{V}}(\mathbf{s}_v, \mathbf{a}_v) = -\omega_L L^{i,j}_v(\mathbf{a}_v, \mathbf{s}_v) - \omega_E E^{i,j}_v(\mathbf{a}_v, \mathbf{s}_v) + \omega_Q Q^{i,j}_v(\mathbf{a}_v, \mathbf{s}_v), \quad (64)$$

where ω_L, ω_E, and ω_Q are the weighting coefficients for latency, energy, and QoS, respectively.

2) UAV Reward Function ($\mathscr{R}_{\mathcal{U}}$): The reward function for UAV agents emphasizes the importance of efficient task forwarding and robust energy management, while also ensuring load balancing. This function is articulated to align UAV actions with the overarching network objectives:

$$\mathscr{R}_{\mathcal{U}}(\mathbf{s}_u, \mathbf{a}_u) = \zeta_{TF} TF^{i,j}_u(\mathbf{a}_u, \mathbf{s}_u) - \zeta_E E^{i,j}_u(\mathbf{a}_u, \mathbf{s}_u), \quad (65)$$

$$TF^{i,j}_u(\mathbf{a}_u, \mathbf{s}_u) = \alpha_\eta \eta^{i,j}_u - \alpha_L L^{i,j}_u + \alpha_{LB} LB^{i,j}_u, \quad (66)$$

$$LB^{i,j}_u = \beta_{TD} TD^{i,j}_u + \beta_{RU} RU^{i,j}_u - \beta_{QL} QL^{i,j}_u, \quad (67)$$

where ζ_{TF} and ζ_E are the weights assigned to task forwarding efficiency and energy consumption, respectively. The task forwarding metric $TF^{i,j}_u$ integrates the forwarding success rate ($\eta^{i,j}_u$), latency ($L^{i,j}_u$), and load balancing ($LB^{i,j}_u$). The load balancing metric $LB^{i,j}_u$ is further defined by the task distribution ($TD^{i,j}_u$), resource utilization ($RU^{i,j}_u$), and queue length ($QL^{i,j}_u$), with respective weights β_{TD}, β_{RU}, and β_{QL}.

3) BS Reward Function ($\mathscr{R}_{\mathcal{B}}$): The reward function for BS agents is designed similarly to the UAV agents. This function aligns BS actions with the network's objectives, focusing on efficient task processing and energy efficiency:

$$\mathscr{R}_{\mathcal{B}}(\mathbf{s}_b, \mathbf{a}_b) = \zeta_{TP} TP^{i,j}_b(\mathbf{a}_b, \mathbf{s}_b) - \zeta_E E^{i,j}_b(\mathbf{a}_b, \mathbf{s}_b), \quad (68)$$

where ζ_{TP} and ζ_E represent the weights assigned to task processing efficiency and energy consumption for the BS agents, respectively. The task processing efficiency metric $TP^{i,j}_b$ and the energy consumption metric $E^{i,j}_b$ are calculated based on the actions and states specific to the BS agents.

V. Proposed Multi-agent LSTM-aided Quantum-DRL Based Solution

In our URLLC-enabled VEC network, each real-world agent is paired with a digital replica within a DT framework. These replicas mirror and predict the states and actions of their corresponding real-world agents, enhancing decision-making through a synchronized learning process. In this setup, LSTM networks are pivotal. The LSTM system integrated into each DT processes historical and current data to predict future states and actions. This foresight is invaluable for optimizing task offloading, energy consumption, and QoS in our network.

A. Enhanced LSTM Network in the DT Framework

The LSTM networks consist of sophisticated units: input gate I, forget gate F, memory cell C, and output gate O. These units are crucial for processing and retaining relevant

historical sequence information, informing future decisions. The standard mathematical representation of the LSTM structure is usually expressed as follows:

$$I(t) = \sigma(W_I o(t) + W_{IH} h(t-1) + W_{IC} C(t-1) + b_I), \quad (69)$$

$$F(t) = \sigma(W_F o(t) + W_{FH} h(t-1) + W_{FC} C(t-1) + b_F), \quad (70)$$

$$C(t) = F(t) \circ C(t-1) + I(t) \circ \tanh(W_C o(t) + W_{CH} h(t-1) + b_C), \quad (71)$$

$$O(t) = \sigma(W_O o(t) + W_{OH} h(t-1) + W_{OC} C(t) + b_O), \quad (72)$$

$$h(t) = O(t) \circ \tanh(C(t)), \quad (73)$$

where $\sigma()$ represents the sigmoid activation function, which maps values into a bounded interval. tanh() denotes the hyperbolic tangent activation function, providing outputs between (-1, 1). $\circ$ symbolizes element-wise (Hadamard) multiplication. $I(t)$, $F(t)$, $C(t)$, and $O(t)$ are the input, forget, memory cell, and output gates at time t, respectively. $h(t)$ refers to the hidden state at time t. The weights associated with these gates are denoted as W_I, W_{IH}, W_{IC} for the input gate; W_F, W_{FH}, W_{FC} for the forget gate; W_C, W_{CH}, W_{CC} for the memory cell; and W_O, W_{OH}, W_{OC} for the output gate. b_I, b_F, b_C, and b_O represent the bias terms for each respective gate.

B. Synchronization with Digital Twins (DTs) and Real World

Each LSTM network is intricately synchronized with its corresponding real-world counterpart in the proposed DT framework. This synchronization enables digital replicas to mirror and predict physical agents' actions and states accurately, thereby enabling a robust learning mechanism using the following loss function:

$$\mathcal{L}_{\text{Loss}} = \sum_{i=1}^{M} (\boldsymbol{o}_i(t) - \hat{\boldsymbol{o}}_i(t))^2 + \eta_w \sum_{i=1}^{M} \text{Var}\left(\hat{\boldsymbol{o}}_i(t)\right), \quad (74)$$

where $\hat{\boldsymbol{o}}(t)$ represents the set of parameters predicted by the LSTM within the DT, including both the state $\mathbf{s} \in \{\mathbf{s}_v, \mathbf{s}_b, \mathbf{s}_u\}$ to its corresponding action $\mathbf{a} \in \{\mathbf{a}_v, \mathbf{a}_b, \mathbf{a}_u\}$ for each agent. $\boldsymbol{o}(t)$ refers to the corresponding actual parameters in the real world at time t. The summation runs over all M parameters of interest, ensuring a comprehensive evaluation. Symbol η_w represents a weighting coefficient that balances the two terms of the loss function.

The present LSTM optimization algorithm in **Algorithm 1** utilizes Adam, a sophisticated optimizer with adaptive learning rates for each parameter, enhancing convergence efficiency. The loss function is augmented with a variance term, encouraging model diversity in predictions and reducing the likelihood of overfitting to the training data.

C. Quantum-DRL in DT and Physical Agents

To effectively address the complexities of the URLLC-enabled VEC network, we propose a quantum-enhanced DRL algorithm in both the DT and physical agents, as shown in Fig. 2. This approach combines the computational potential of quantum computing with the adaptive learning abilities of DRL. Quantum computing operates on the principles of quantum mechanics [31], utilizing qubits that can exist in superposition and entanglement states [32], [33]. This enables quantum computers to process a vast amount of data simultaneously, offering significant computational advantages over classical computing, especially in optimizing the present complex scenario of vehicular task offloading. To mathematically derive and analyze the integration of quantum computing in DRL, particularly for a Quantum-DRL model applied to the present work, we need to explore several key components: state encoding, quantum policy network, and action selection. We frame each of them as follows:

Algorithm 1 LSTM Optimization for DT Synchronization.

Initialization:

Learning rate: $\alpha = 0.001$.
Adam parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.
Number of epochs: N, Batch size: B.
Initialize LSTM parameters $\boldsymbol{\theta}_{\text{LSTM}} \sim \mathcal{N}(0, 1/n)$.
Initialize moment vectors: $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$.

Optimization Procedure:

for $epoch = 1$ to N **do**
  for each batch **B** from dataset **do**
    Compute gradients $\nabla\boldsymbol{\theta}_{\text{LSTM}}$ using BPTT:
    $\nabla\boldsymbol{\theta}_{\text{LSTM}} = \frac{\partial \mathcal{L}_{\text{Loss}}}{\partial \boldsymbol{\theta}_{\text{LSTM}}}$.
    Update moment vectors:
    $\mathbf{m}_{epoch} = \beta_1 \mathbf{m}_{epoch-1} + (1-\beta_1)\nabla\boldsymbol{\theta}_{\text{LSTM}}$.
    $\mathbf{v}_{epoch} = \beta_2 \mathbf{v}_{epoch-1} + (1-\beta_2)\nabla\boldsymbol{\theta}^2_{\text{LSTM}}$.
    Correct bias:
    $\hat{\mathbf{m}}_{epoch} = \frac{\mathbf{m}_{epoch}}{1-\beta_1^{epoch}}$.
    $\hat{\mathbf{v}}_{epoch} = \frac{\mathbf{v}_{epoch}}{1-\beta_2^{epoch}}$.
    Update LSTM parameters:
    $\boldsymbol{\theta}_{\text{LSTM}} = \boldsymbol{\theta}_{\text{LSTM}} - \alpha \frac{\hat{\mathbf{m}}_{epoch}}{\sqrt{\hat{\mathbf{v}}_{epoch}+\epsilon}}$.
  end for
  Evaluate model on validation set.
  Adjust learning rate if needed.
end for
return Optimized parameters $\boldsymbol{\theta}_{\text{LSTM}}$.

1) Classical State Representation: Let us denote the classical state of the network at time t as $\mathbf{s}(t)$, which includes all relevant information about vehicles (v), UAVs (u), and BS (b). This can be expressed as a vector: $\mathbf{s}(t) = [\mathbf{s}_v(t), \mathbf{s}_u(t), \mathbf{s}_b(t)]$, where each $\mathbf{s}_x(t)$ represents the state of component x in the network.

2) Quantum State Encoding: In our model, quantum states are encoded from classical states via an encoding function $\mathcal{E}$, crucial for the multi-dimensional representation in quantum computation. For a quantum system with N qubits, the state $|\psi\rangle$ is a superposition of basis states, formulated as [18] $|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle$, where α_i are complex coefficients fulfilling normalization. The encoding function $\mathcal{E}$ transforms classical state vectors $\mathbf{s}(t)$ into quantum states in a 2^N-dimensional Hilbert space, a process critical for leveraging quantum computing in DRL.

3) Quantum Encoding Process: Given a classical state vector $\mathbf{s}(t)$ in a real or discrete space, the encoding function $\mathcal{E}$ maps this state to a quantum state $|\psi(t)\rangle$ in a 2^N-dimensional Hilbert space, defined as:

$$|\psi(t)\rangle = \mathcal{E}(\mathbf{s}(t)) = \sum_{i=0}^{2^N-1} \alpha_i(\mathbf{s}(t))|i\rangle. \quad (75)$$

The coefficients $\alpha_i(\mathbf{s}(t))$ are complex-valued functions of $\mathbf{s}(t)$, derived from a set of basis functions $\{f_i(\mathbf{s}(t))\}_{i=0}^{2^N-1}$, and are normalized as:

$$\alpha_i(\mathbf{s}(t)) = \frac{f_i(\mathbf{s}(t))}{\sqrt{\sum_{j=0}^{2^N-1} |f_j(\mathbf{s}(t))|^2}}, \tag{76}$$

subject to the condition $\sum_{i=0}^{2^N-1} |\alpha_i(\mathbf{s}(t))|^2 = 1$.

Lemma 1. *Let $\mathbf{s}(t)$ be a classical state vector and $\mathcal{E}$ an encoding function for an N-qubit system. If for every $\mathbf{s}(t)$, the set of basis functions $\{f_i\}$ satisfies $\sum_{i=0}^{2^N-1} |f_i(\mathbf{s}(t))|^2 > 0$, then the quantum state $|\psi(t)\rangle = \mathcal{E}(\mathbf{s}(t))$ is well-defined and normalized.*

Proof. Given the basis functions $\{f_i(\mathbf{s}(t))\}$, the coefficients $\alpha_i(\mathbf{s}(t))$ are computed by:

$$\alpha_i(\mathbf{s}(t)) = \frac{f_i(\mathbf{s}(t))}{\sqrt{\sum_{j=0}^{2^N-1} |f_j(\mathbf{s}(t))|^2}}. \tag{77}$$

To satisfy the normalization condition $\sum_{i=0}^{2^N-1} |\alpha_i(\mathbf{s}(t))|^2 = 1$, we have:

$$\sum_{i=0}^{2^N-1} \left| \frac{f_i(\mathbf{s}(t))}{\sqrt{\sum_{j=0}^{2^N-1} |f_j(\mathbf{s}(t))|^2}} \right|^2 = \frac{\sum_{i=0}^{2^N-1} |f_i(\mathbf{s}(t))|^2}{\sum_{j=0}^{2^N-1} |f_j(\mathbf{s}(t))|^2} = 1. \tag{78}$$

Hence, $|\psi(t)\rangle$ is a valid quantum state. □

4) Quantum Policy Network: The quantum policy network ($\mathcal{QPN}$) then inputs these quantum states and outputs transformed states for decision-making. Specifically, the network uses a quantum circuit $\mathcal{Q}$, parameterized by unitary transformations, to map encoded states to a probabilistic action space, facilitating optimal action selection under network constraints,

$$|\psi(t)\rangle = \mathcal{E}(\mathbf{s}(t)) = \sum_{i=0}^{2^N-1} \alpha_i(\mathbf{s}(t))|i\rangle, \tag{79}$$

$$|\phi(t)\rangle = \mathcal{Q}(|\psi(t)\rangle) = U(\boldsymbol{\theta}, t)|\psi(t)\rangle. \tag{80}$$

In the $\mathcal{QPN}$, the core mechanisms of quantum computing, namely superposition and entanglement, are crucial for its functionality. Superposition allows the network to consider multiple possible states simultaneously rather than being limited to a single state at any given time. This dramatically expands the network's capacity to explore many potential actions within its action space. Entanglement, on the other hand, enables a level of interaction between quantum states. This interaction is pivotal for the $\mathcal{QPN}$ as it allows for complex relationships and dependencies between different states, essential in making well-informed decisions.

The operations within the $\mathcal{QPN}$ are orchestrated by a unitary matrix, denoted as $U(\boldsymbol{\theta}, t)$. This matrix is a mathematical representation of a sequence of quantum gates. Each quantum gate in this sequence performs a specific transformation on the network's quantum states, and their composition, dictated by $U(\boldsymbol{\theta}, t)$, is what executes the decision-making process. The parameterization of this matrix by $\boldsymbol{\theta}$ and its potential time-dependence, t, allow for a dynamic and adaptable system capable of handling a variety of scenarios and decision criteria.

a) Quantum Circuit Representation: The quantum circuit $\mathcal{Q}$, operating on an N-qubit system, acts on the input quantum state $|\psi(t)\rangle$ as follows:

$$|\phi(t)\rangle = \mathcal{Q}(|\psi(t)\rangle) = U(\boldsymbol{\theta}, t)|\psi(t)\rangle. \tag{81}$$

Here, $U(\boldsymbol{\theta}, t)$ is a time-dependent unitary matrix, parameterized by $\boldsymbol{\theta}$, which represents the gate parameters of the quantum circuit.

b) Decomposition of Unitary Matrix: The unitary matrix U is decomposed into a sequence of quantum gates, each performing single-qubit or multi-qubit operations:

$$U(\boldsymbol{\theta}, t) = U_N(\theta_N, t) s U_2(\theta_2, t) U_1(\theta_1, t), \tag{82}$$

where each $U_i(\theta_i, t)$ signifies a unitary operation corresponding to the i-th gate, controlled by parameter θ_i.

c) Gate Decomposition and Quantum Evolution: The evolution of the quantum state is governed by the Hamiltonian $H(\boldsymbol{\theta})$ of the system, leading to:

$$U(\boldsymbol{\theta}, t) = e^{-iH(\boldsymbol{\theta})t/\hbar}, \tag{83}$$

where $H(\boldsymbol{\theta})$ represents a sum of Hamiltonians for each gate, and $\hbar$ is the reduced Planck constant.

d) Entanglement and Superposition: Entanglement and superposition are critical in the quantum policy network. Unitary operations induce entanglement, allowing for the exploration of a complex state space, and superposition enables simultaneous evaluation of multiple policy outcomes. Entangled states are represented as:

$$|\psi_{\text{entangled}}\rangle = \sum_{i,j} \beta_{ij} |i\rangle \otimes |j\rangle, \tag{84}$$

where β_{ij} coefficients signify entanglement, not factorizable into products of individual state probabilities.

e) Quantum Measurement and Policy Decision: The quantum state $|\phi(t)\rangle$ undergoes a measurement process, collapsing to a classical outcome for policy decisions. This process is characterized by:

$$P(m) = \langle \phi(t) | P_m | \phi(t) \rangle, \tag{85}$$

where $\{P_m\}$ are projection operators, and $P(m)$ is the probability of observing outcome m, forming the basis of decision-making in our quantum-enhanced DRL model.

5) Action Selection in Quantum Multi-Agent DRL Framework: The action selection in our Quantum-DRL framework is a critical process intricately designed to handle the multi-agent dynamics of the URLLC-enabled VEC network. This process integrates quantum computing principles with DRL techniques to efficiently navigate the complexities of the problem space, aiming to optimize network performance while adhering to constraints like latency, energy efficiency, and QoS.

a) Quantum Multi-Agent Optimization: In our approach, each agent (vehicle, UAV, and BS) independently selects actions that contribute to the global objective of the network. The optimal action set $\mathbf{a}^*(t)$ at time t for all agents is determined by collaboratively solving an optimization problem in (38):

$$\mathbf{a}^*(t) = \arg\min_{\mathbf{a} \in \mathcal{A}} \mathcal{J}(\mathbf{s}(t), \mathbf{a}), \tag{86}$$

where $\mathcal{J}$ is the objective function encapsulating latency, energy, and QoS metrics as mentioned in (38).

Algorithm 2 Multi-Agent Quantum-DRL to Solve Eq. (38).

Initialize: Quantum policy networks $\Pi_{\theta_v}, \Pi_{\theta_u}, \Pi_{\theta_b}$ for each agent type.
Initialize critic $C_{\phi_v}, C_{\phi_u}, C_{\phi_b}$, experience replay $\mathcal{D}_v, \mathcal{D}_u, \mathcal{D}_b$.
Set learning rates α_Π, α_C, discount factor γ, and exploration rate ϵ.
for each episode $e = 1, 2, \ldots, E$ **do**
  Reset environment and obtain initial state $\mathbf{s}(0)$.
  for each time step $t = 1, 2, \ldots, T$ **do**
    for each agent type $x \in \{v, u, b\}$ **do**
      Encode classical state $\mathbf{s}_x(t)$ into quantum state $|\psi_x(t)\rangle$.
      Compute reward $\mathscr{R}_x$ for state $\mathbf{s}_x(t)$ using **Algorithm 3**.
      Select action $\mathbf{a}_x(t)$ using Π_{θ_x} with exploration and $\mathscr{R}_x$.
      Execute $\mathbf{a}_x(t)$ and find reward $r_x(t)$ and new state $\mathbf{s}_x(t+1)$.
      Store transition $(\mathbf{s}_x(t), \mathbf{a}_x(t), r_x(t), \mathbf{s}_x(t+1))$ in $\mathcal{D}_x$.
      Sample random minibatch from $\mathcal{D}_x$.
      Update critic C_{ϕ_x} by minimizing loss:

$$\mathcal{L}(\phi_x) = \mathbb{E}_{\mathcal{D}_x}\left[\left(r_x(t) + \gamma C_{\phi'_x}(\mathbf{s}_x(t+1)) - C_{\phi_x}(\mathbf{s}_x(t))\right)^2\right].$$

      Update policy Π_{θ_x} using gradient ascent:

$$\nabla_{\theta_x} J(\Pi_{\theta_x}) = \mathbb{E}_{\mathcal{D}_x}\left[\nabla_{a_x} Q(\mathbf{s}_x, \mathbf{a}_x|\phi_x)|_{\mathbf{a}_x=\Pi_{\theta_x}(\mathbf{s}_x)} \nabla_{\theta_x} \Pi_{\theta_x}(\mathbf{s}_x)\right].$$

    end for
    if terminal state or $t = T$ **then**
      Break and proceed to the next episode.
    end if
  end for
end for

b) Quantum-Assisted Decision Making: Each agent employs quantum computing techniques to explore the vast action space efficiently. This is achieved by encoding the decision-making problem into a quantum state and utilizing quantum algorithms for optimization:

$$|\psi_{\text{opt},x}(t)\rangle = \arg\min_{|\psi_x\rangle} \langle\psi_x|\hat{\mathcal{J}}_x(\mathbf{s}_x(t), \hat{\mathbf{a}}_x)|\psi_x\rangle. \tag{87}$$

Here, $|\psi_{\text{opt},x}(t)\rangle$ represents the quantum state corresponding to the optimal action for agent type x, and $\hat{\mathcal{J}}_x$ and $\hat{\mathbf{a}}_x$ denote the quantum representations of the objective function and action operators, respectively.

c) Quantum Policy Network Integration: In the Quantum-DRL model, $\mathcal{QPN}$s are integral for each agent type, formalized mathematically as:

$$\Pi_{\mathcal{QPN}}(|\psi(t)\rangle) = \sum_{a\in\mathcal{A}} p(a|\psi(t))a, \tag{88}$$

where $|\psi(t)\rangle$ is the quantum-encoded state of the network at time t, and $\mathcal{A}$ is the action space. The function $\Pi_{\mathcal{QPN}}$ maps $|\psi(t)\rangle$ to a probability distribution over the action space, with $p(a|\psi(t))$ representing the probability of choosing action a given the state $|\psi(t)\rangle$. This mapping is crucial for action selection and alignment to optimize global network objectives while adhering to the constraints of the URLLC-enabled VEC environment. The training of $\mathcal{QPN}$ involves adjusting its parameters to maximize the expected reward, given by:

$$\mathbb{E}_{\Pi_{\mathcal{QPN}}}[\mathscr{R}] = \sum_{\mathbf{s}\in\mathcal{S}} \rho(\mathbf{s}) \sum_{a\in\mathcal{A}} p(a|\psi(\mathbf{s}))\mathscr{R}(\mathbf{s}, a), \tag{89}$$

where $\mathscr{R}(\mathbf{s}, a)$ is the reward function, $\rho(\mathbf{s})$ is the distribution over states, and $\mathcal{S}$ is the set of all states. Through this quantum-assisted multi-agent DRL approach, encapsulated in **Algorithm 2**, our model adeptly balances individual agent autonomy with global network optimization, effectively tackling the challenges in task offloading, resource allocation, and enhancing overall network performance in URLLC-enabled VEC networks.

D. Incentive and Penalty Functions in Reward Optimization

In our multi-agent Quantum-DRL framework, we introduce incentive and penalty functions to encourage desirable actions and discourage actions that could negatively impact the network's performance. These functions are integrated into the Nash equilibrium-based reward optimization algorithm (**Algorithm 3**), as outlined in Section 3.

1) Incentive Function: The incentive function, $\mathcal{I}_x(\mathbf{a}_x, \mathbf{s}_x)$, rewards agents for actions that align with the network's objectives, such as reducing latency, improving energy efficiency, and enhancing QoS. It is defined as a weighted sum of the following performance incentive metrics:

- Latency reduction incentive: We use an incentive function to reward the agent for reducing latency as follows:

$$\mathcal{I}_{\text{lat}}(\mathbf{a}_x, \mathbf{s}_x) = (\omega_{\text{lat}}) \max(0, \varrho - L_x(\mathbf{a}_x, \mathbf{s}_x)), \tag{90}$$

  where $0 < \omega_{\text{lat}} < 1$ is the weight for latency reduction, ϱ is the target latency threshold, and $L_x(\mathbf{a}_x, \mathbf{s}_x)$ represents the latency for action $\mathbf{a}_x$ and state $\mathbf{s}_x$.
- Energy efficiency incentive: The following function encourages the agent for energy-efficient actions:

$$\mathcal{I}_{\text{ene}}(\mathbf{a}_x, \mathbf{s}_x) = \omega_{\text{ene}}\left(1 - \frac{E_x(\mathbf{a}_x, \mathbf{s}_x)}{E^x_{\max}}\right), \tag{91}$$

  where the weight incentive is $0 < \omega_{\text{ene}} < \omega_{\text{lat}}$, and $E_x(\mathbf{a}_x, \mathbf{s}_x)$ is the energy used by the agent for the given action and state, and $E^x_{\max}$ is the maximum allowed energy consumption.
- QoS incentive: The following function provides an incentive in reward calculation for improving the QoS:

$$\mathcal{I}_{\text{qos}}(\mathbf{a}_x, \mathbf{s}_x) = \omega_{\text{qos}} \max(0, Q_x(\mathbf{a}_x, \mathbf{s}_x) - Q_{\min}), \tag{92}$$

  where $0 < \omega_{\text{qos}} < 1 - (\omega_{\text{lat}} + \omega_{\text{ene}})$, and $Q_x(\mathbf{a}_x, \mathbf{s}_x)$ measures the QoS using (48) for the action and state, with $Q_{\min}$ being the minimum QoS target.

2) Penalty Function: The penalty function, $\mathcal{P}_x(\mathbf{a}_x, \mathbf{s}_x)$, imposes penalties on actions that deviate from the network's objectives or violate constraints. It is defined as a weighted sum of the following defined penalty metrics:

- Latency excess penalty: The function $\mathcal{P}_{\text{lat}}(\mathbf{a}_x, \mathbf{s}_x) = \lambda_{\text{lat}} \max(0, L_x(\mathbf{a}_x, \mathbf{s}_x) - \varrho)$ penalizes exceeding the latency threshold. We consider $\varrho = 10$ ms for this work. $0 < \lambda_{\text{lat}} < 1$ is the penalty weight for latency excess.
- Energy excess penalty: Function $\mathcal{P}_{\text{ene}}(\mathbf{a}_x, \mathbf{s}_x) = \lambda_{\text{ene}} \max(0, E_x(\mathbf{a}_x, \mathbf{s}_x) - E^x_{\max})$ penalizes for exceeding the maximum allowed energy consumption. λ_{ene} is the penalty weight and $0 < \lambda_{\text{ene}} < \lambda_{\text{lat}}$.
- QoS deficit penalty: Function $\mathcal{P}_{\text{qos}}(\mathbf{a}_x, \mathbf{s}_x) = \lambda_{\text{qos}} \max(0, Q_{\min} - Q_x(\mathbf{a}_x, \mathbf{s}_x))$ penalizes for falling below the minimum QoS target. λ_{qos} is the penalty weight and $0 < \lambda_{\text{qos}} < 1 - (\lambda_{\text{ene}} + \lambda_{\text{lat}})$.

These functions are used within the Nash equilibrium-based reward optimization

TABLE II: Simulation Parameters for the Proposed Work.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$f^z_{\text{cpu}}(t)$	10 GHz [14]	σ_z^2	-90 dB	$P_{\max,v}$	-30 dB [29]	$\widehat{R}(t)$	10 Mbps [4]
U	10	$\overline{f^z_{\text{cpu}}(t)}$	± 0.15 GHz [4]	$P_{\max,b}$	-10 dB [4]	τ	20 ms [4]
B	4	$\hat{T}$	1 s	$P_{\max,u}$	0 dB	δ	0.005 m [4]
θ_v	$(0, 2\pi)$	ϕ_v	$(-10, 10)^\circ$	M	4	l_x	100
α_{PL}	(2.25, 2.75)	ϵ_b	10^{-6}	$\mathscr{C}_z$	2.7×10^8	ϱ	10 ms [4]
$E^v_{\max}$	0.5 Joule [4]	$\mathscr{B}$	10 MHz [14]	$Q_{\min}$	60	$\Delta S_{\max}$	0.5
ξ_z	10^{-6} J/cycle	$D^{\max}_{i,j}$	1 MB	$E^z_{\max}$	33.2 J	$C_{i,j}$	330/byte [4]
λ	0.009 m	c	3e+8 m/s [4]	$\widehat{V_u}$	(10, 50) m/s	$\widehat{V_v}$	20 m/s [29]

Algorithm 3 Nash Equilibrium Based Reward Optimization.

Input: $\mathcal{S}_v$, $\mathcal{S}_u$, $\mathcal{S}_b$; $\mathcal{A}_v$, $\mathcal{A}_u$, $\mathcal{A}_b$; Π_{θ_v}, Π_{θ_u}, Π_{θ_b}.
Initialize:
for each agent type $x \in \{v, u, b\}$ **do**
  Initialize reward functions $\mathscr{R}_x$.
  Initialize Quantum-DRL policies Π_{θ_x}.
  Initialize constraint functions C_x.
end for
Nash Equilibrium Computation:
while not converged **do**
  Define utility functions $U_x(\mathbf{a}_x, \mathbf{a}_{-x})$ for each agent x.
  Compute Nash equilibrium $\mathbf{a}^*$ satisfying:
  for each agent x **do**
    $\mathbf{a}^*_x = \arg\max_{\mathbf{a}_x \in \mathcal{A}_x} U_x(\mathbf{a}_x, \mathbf{a}^*_{-x})$.
  end for
  Update Π_{θ_x} based on $\mathbf{a}^*$ for each agent x.
end while
Reward Adjustment with Incentives and Penalties:
for each agent x at each time step t **do**
  Compute incentive $\mathcal{I}_x(\mathbf{a}_x, \mathbf{s}_x)$ based on defined incentive functions.
  Compute penalty $\mathcal{P}_x(\mathbf{a}_x, \mathbf{s}_x)$ based on defined penalty functions.
  Adjusted reward $\mathscr{R}^{adj}_x = \mathscr{R}_x + \mathcal{I}_x - \mathcal{P}_x$.
end for

E. Handling Network Disruptions

Our framework employs Quantum-DRL and LSTM within a DT architecture to ensure stability against network disruptions and data inconsistencies. Quantum-DRL dynamically adapts to changing network conditions, while LSTM predicts future states, enhancing the system's preemptive adjustment capabilities. The DT simulates various disruption scenarios, allowing for strategy testing and preparation. Integrated robustness, redundancy, and edge computing further strengthen the system, enabling local data processing and reducing central server dependency. This multi-layered approach ensures uninterrupted operation and reliability, crucial for real-world vehicular network applications.

VI. Numerical Results and Analysis

In our system, the decision-making process, powered by LSTM and Quantum-DRL algorithms, is centrally executed within the DT in the cloud. This setup includes digital agents consistently synchronized with physical world agents in BS and UAVs through a dedicated feedback channel. While computing in the cloud-based DT, the Quantum-DRL algorithms directly influence the physical agents' task processing and operational strategies, ensuring efficient network management. In our experimental setup, the Quantum-DRL network is configured with 4 quantum circuit layers, each comprising 6 qubits to represent the state space intricately. A combination of Pauli_X, Pauli_Y, Pauli_Z, Hadamard, and CNOT gates, alongside parameterized rotation gates (R_X, R_Y, R_Z), is utilized for quantum operations. Google's Cirq framework and TensorFlow Quantum are used to construct and simulate quantum circuits. The LSTM networks in DT incorporate 3 layers, each with 128 neurons, a dropout rate of 0.3, and use the ReLU activation function. Training is conducted over 1000 epochs with a batch size of 64 using the Adam optimizer at a learning rate of 0.001. The DRL model employs an epsilon-greedy exploration strategy with a carefully designed reward function focusing on latency, energy efficiency, and QoS. The discount factor γ is set to 0.99, emphasizing the significance of future rewards in the agent's decision-making process.

In our simulation, BSs are evenly distributed across a 3D space, with x and y coordinates ranging from -1000 to 1000 meters and a z-axis height of 40 to 60 meters. UAVs are allocated within a broader area of -1500 to 1500 meters on both x and y axes and at heights of 30 to 75 meters. Following the RWP model, vehicles move in a space extending from -1250 to 1250 meters on both x and y axes at altitudes between 0 and 20 meters. The rest of the simulation parameters and their respective values are listed in Table II, where symbol x and y present $x \in \{\mathcal{B} \cup \mathcal{U} \cup \mathcal{V}\}$ and $z \in \{\mathcal{B} \cup \mathcal{U}\}$, respectively. We select some crucial parameter values from [4], [14], [29] while other parameter values are fixed carefully to fit practically in our model.

As depicted in Fig. 3, the proposed Quantum-DRL algorithm significantly outperforms the standard PPO [13] and DDPG [12] algorithms where all the algorithms are executed in a multi-agent environment along with LSTM. At the 1000^{th} episode, Quantum-DRL demonstrates a substantial performance gain, accruing a normalized reward value that surpasses PPO's by approximately 53.17% and DDPG's by 84.81%. This pronounced improvement is particularly evident beyond episode 400, where Quantum-DRL consistently maintains an elevated reward value. This trend culminates in near convergence around episode 600, marking the stabilization of policy learning. Contrary to this, PPO's convergence trajectory is more gradual, with a noticeable stabilization near episode 800. As seen in Fig. 3, the multi-agent Quantum-DRL algorithm shows performance similar to the Standard multi-agent PPO Algorithm around episode 200. This observation is indeed correct and reflects Quantum-DRL's exploration phase. It is important to note that during this phase, the rewards might not always reflect the algorithm's ultimate capability as it is

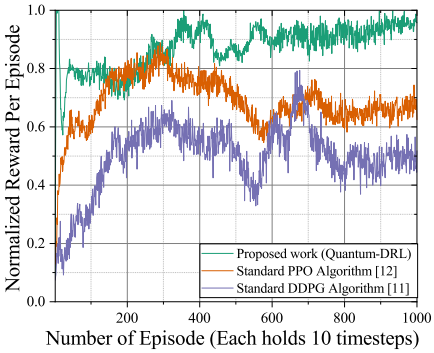

Fig. 3: Convergence plot.

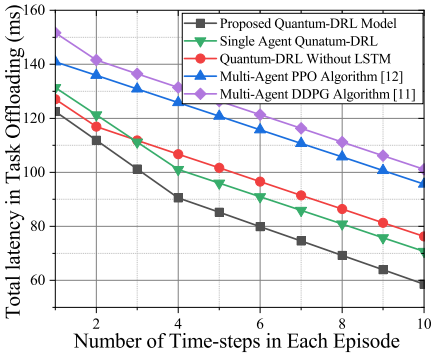

Fig. 4: Impact of DRL time-steps in latency.

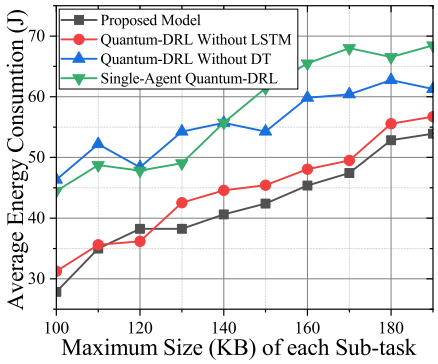

Fig. 5: Analysis of Sub-task size effects.

not yet focused on exploiting the learned strategies. As the episodes progress, Quantum-DRL begins to utilize its learned experiences, transitioning more towards exploitation, which leads to an increase in reward and a reduction in the variance of the rewards. This transition is evidenced by the higher and more stable rewards obtained from around episode 600 onwards, surpassing the standard multi-agent PPO and DDPG.

In task offloading latency in the proposed work, we consider the actual task (i.e., $\mathcal{T}_i$) size to be 1 MB. Fig. 4 plots total latency (i.e., network plus processing) versus the number of time steps in each episode. In our experimental evaluation, the proposed multi-agent-Quantum-DRL with LSTM in the DT model demonstrated superior performance over its counterparts. At the 10^{th} time step, it achieved a significant reduction in latency, outperforming the Single-Agent Quantum-DRL by approximately 30.11%, the proposed multi-agent-Quantum-DRL without LSTM by about 63.18%, the multi-agent PPO Model by roughly 20.72%, and the multi-agent DDPG model by approximately 72.47%. The primary reasons for this notable performance include the advanced predictive capabilities of the LSTM integrated within the DT, which enabled more accurate forecasting and adaptation to the dynamic network environment. As we can note from Fig. 4, the proposed multi-agent Quantum-DRL model significantly outperforms standard multi-agent PPO and DDPG algorithms in minimizing task offloading latency across various timesteps. This is clear from the consistently lower latency achieved by multi-agent and single-agent Quantum-DRL configurations. Interestingly, even when the LSTM component is removed from the Quantum-DRL model, it performs better than its LSTM-equipped PPO and DDPG models. This highlights the robustness of the Quantum-DRL and its superior ability to adapt to dynamic environments. Including LSTM further enhances this capability, as shown by the increased latency in the non-LSTM version, particularly as the number of timesteps grows. This underlines the critical role of LSTM in capturing temporal dependencies essential for efficient policy exploitation over time.

Fig. 5 demonstrates the superior energy efficiency of the proposed Quantum-DRL model with LSTM for vehicular network task offloading. At a maximum sub-task size of 100 KB, the model reduces energy consumption by 10.83% compared to the Quantum-DRL without LSTM, by 39.82% relative to Quantum-DRL without DT, and by 37.30% against the Single

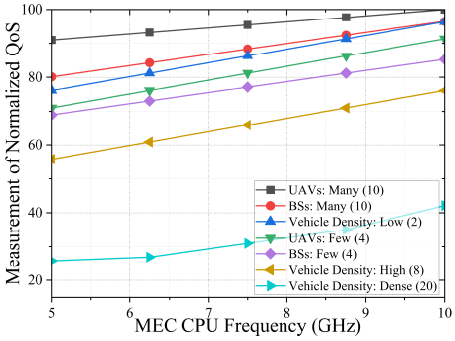

Fig. 6: Improve QoS by varying CPU frequencies in diverse scenarios.

Agent Quantum-DRL. This performance enhancement is primarily due to the integration of LSTM within the DT framework, which enables predictive modeling of network conditions, thereby facilitating more efficient offloading decisions and reducing energy consumption. The use of LSTM becomes particularly beneficial as the task size increases, demanding the transmission of more data packets under stringent network conditions. Moreover, the model optimizes energy use more effectively than single-agent systems, which may incur higher penalties and achieve suboptimal results due to centralized decision-making in complex network environments. Including DT in the multi-agent setup ensures better synchronization between actual and virtual replicas of the network, optimizing sub-task distribution and enhancing overall energy efficiency.

To effectively represent and analyze the QoS values $\forall\{i,j\}$ within the network, as illustrated in Fig. 6, we implement a normalization process for the QoS constraint outlined in $C10$. The adjustment is calculated as follows:

$$Q_{i,j}(t) = 100 \times \left(\frac{\frac{w_L}{L_{i,j}(t)} - w_E E_v^{i,j}(t) - Q_{\min}}{Q_{\max} - Q_{\min}} \right) + Q_{\min}, \quad (93)$$

where $w_L = 0.7$ and $w_E = 0.3$, emphasizing the greater importance of reducing latency (w_L) due to its critical impact on QoS, compared to energy expenditure (w_E). This weighting reflects our prioritization of latency minimization in the network's performance criteria. Further analysis reveals that adding MEC nodes, including BSs and UAVs, enhances the network's QoS by exploiting increased computational capabilities. However, it is important to note that as the number

of vehicles in the network increases, a reduction in QoS occurs. This decline is primarily due to the increased demand for task offloading, which exhausts the processing capacities of the MEC nodes. Particularly, the simultaneous handling of multiple requests by the BSs and UAVs reduces the available CPU resources, $f^b_{\text{cpu}}(t)$ and $f^u_{\text{cpu}}(t)$ respectively, thereby impacting the QoS negatively. This scenario highlights the challenges in maintaining high QoS in densely populated vehicle environments and the need for efficient resource management strategies to optimize network performance. In our analysis of MEC processing speeds at 10 GHz, the effectiveness of utilizing UAVs over BSs for improving network QoS becomes evident. For instance, with a high concentration of UAVs and a lower vehicle density, QoS reaches an optimal value of 100, in contrast to a scenario with fewer UAVs and a higher vehicle density, where the QoS is 85.25, marking a significant improvement of approximately 17.30%. This performance enhancement is attributed to the advantageous LoS conditions afforded by UAVs, typically deployed at heights ranging from 22.5 to 100 meters. Furthermore, the LoS conditions are favourable in UAV scenarios compared to the NLoS conditions in BS communications, which suffer from Rayleigh fading over average distance $\approx$ 1000 meters, resulting in higher latency and reduced QoS.

Furthermore, as shown in Fig. 6, indicate that when the computing capability of each MEC node is set at 5 GHz, the dense vehicular network (with 20 vehicles) experiences the lowest QoS as when the number of BSs and UAVs is 4. However, as we incrementally increase the processing capability of MEC, there is a corresponding improvement in QoS. This demonstrates that while our system can handle more vehicles, it also requires sufficient network resources to manage the increase in task offloading demands effectively.

Fig. 6 demonstrates a distinct dependency on CPU frequency in QoS performance during task offloading. For example, at a CPU frequency of 7.5 GHz and a low vehicle density (2), the system achieves a QoS of 86.26, suggesting efficient handling in less congested environments. However, increasing vehicle density to 8 causes a reduction in QoS to 65.96, highlighting congestion's detrimental effects on service quality. In scenarios with a limited number of UAVs (4), the QoS slightly improves to 87.27, indicating the beneficial role of UAVs in offloading processes. Contrarily, a larger UAV count (10) leads to a lower QoS of 77.12, pointing to potential inefficiencies from managing more nodes. Similarly, a sparse BS setup (4) yields a QoS of 81.18, reflecting respectable performance with minimal infrastructure. Remarkably, an increased number of BSs (10) boosts the QoS to 97.42, highlighting the importance of infrastructure adequacy for optimal network service.

Fig. 7 illustrates the relationship between network latency and finite block length (FBL) in URLLC services. The analysis reveals a clear trend of increasing latency with larger FBLs across various scenarios involving different vehicle densities $\in (2, 8)$, UAV numbers $\in (4, 10)$, and BS distributions $\in (4, 10)$. In a low-density traffic environment, a UAV can connect with a maximum of two vehicles and four BSs concurrently. On the other hand, in a high-density traffic environment, a UAV can connect with up to ten BSs and eight vehicles. For instance,

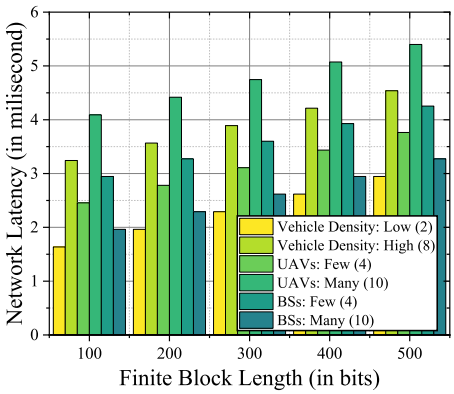

Fig. 7: Network latency against block length for various scenarios.

TABLE III: Analysis of different path-loss models for the vehicle.

	Latency: Vehicle-to-BS		Latency: Vehicle-to-UAV	
Task Size	Okumura	3GPP UMa	Free Space	3GPP UMa
100 KB	0.001010 s	0.008986 s	0.0018 s	0.0018 s
1 MB	0.010109 s	0.089863 s	0.0184 s	0.0180 s
10 MB	0.101093 s	0.898631 s	0.1841 s	0.1805 s

at a block length of 500 bits, the latency in a low vehicle density environment is observed at 2.94 ms, representing a substantial increase compared to the latency at 100 bits, recorded at 1.63 ms. This increase in latency, a consequence of extended encoding and decoding times inherent in longer FBLs, is particularly pronounced in high-density vehicular environments and with greater numbers of UAVs and BSs. The data highlights the criticality of optimizing FBL in URLLC systems to achieve an effective balance between throughput and latency, adhering to the theoretical framework where latency is proportionally related to the block length (n), i.e., $L(n) \propto n$.

Our comprehensive evaluation compares various path-loss models, investigating their impact on latency during vehicular network data transmissions, as listed in Table III. We evaluate the Okumura model [23] against the 3GPP UMa path-loss model (see (16)-(20)) [24] for vehicle-to-BS communications and compare the free space path-loss model with the 3GPP UMa model [24] for vehicle-to-UAV links (see (27)-(29)). Present findings indicate that the Okumura model consistently outperforms the 3GPP UMa model in predicting lower latencies across varied data sizes for vehicle-to-BS links due to its lower path loss estimations in urban scenarios, deviating from the 3GPP UMa model's conservative design for diverse urban conditions. For the present simulation, we consider $d'_{\text{BP}} = 100$ m, $h_{\text{BS}} = 25$ m, $h_{\text{UT}} = 1.5$ m, and block length $n = 1000$. Contrarily, vehicle-to-UAV communication latencies exhibit minimal differences between the free space and 3GPP UMa models, indicating both models' convergence in LoS conditions typical of UAV communications, where the free space model's idealistic clear LoS assumptions closely match the real-world scenarios captured by the 3GPP UMa model's LoS considerations.

Our analysis highlights three prime factors used in our proposed algorithm: (i) Transitioning to a multi-agent setup from a single-agent framework significantly enhances perfor-

mance through distributed cooperative optimization based on Nash equilibrium, as demonstrated in Fig. 4. (ii) Quantum computing integration facilitates parallel computation for state space exploration and action selection, offering an edge over conventional PPO and DDPG-based DRL algorithms by reducing training episodes and time, as shown in Fig. 3, addressing a significant limitation of traditional DRL methods. (iii) Incorporating LSTM strategies adds predictive analysis based on historical data, enhancing decision accuracy by predicting the future states, validated through numerical analysis in Fig. 5.

VII. Conclusions

In this study, we developed an innovative framework for task offloading in URLLC-enabled VEC networks, utilizing a multi-agent system that integrates quantum-enhanced DRL with LSTM networks within a DT architecture. Our Quantum-DRL algorithm surpassed traditional DRL methods, achieving a 53.17% and 84.81% enhancement over standard PPO and DDPG, respectively in convergence and reward maximization. Further, integrating LSTM within the DT significantly improved predictive analytics, leading to efficient decision-making in task offloading, energy management, and QoS. This optimization resulted in a 30.11% reduction in task offloading latency compared to multi-agent-Quantum-DRL without LSTM and a 39.82% decrease in energy consumption against the multi-agent-Quantum-DRL without DT. This research emphasizes the effectiveness of combining quantum computing with advanced machine learning techniques in a distributed technology environment, opening up possibilities for improved optimization of future ITS networks.

References

[1] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8810–8824, Aug. 2022.

[2] Y. Cui *et al.*, "DriveLLM: Charting the path toward full autonomous driving with large language models," *IEEE Trans. Intell. Veh.*, pp. 1–15, 2023.

[3] Y. Chen, Y. Pan, and D. Dong, "Quantum language model with entanglement embedding for question answering," *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 3467–3478, Jun. 2023.

[4] A. Paul, K. Singh, M.-H. T. Nguyen, C. Pan, and C.-P. Li, "Digital twin-assisted space-air-ground integrated networks for vehicular edge computing," *IEEE J. Sel. Top. Signal Process.*, pp. 1–16, 2023.

[5] H. Zhong, L. Wang, J. Cui, J. Zhang, and I. Bolodurina, "Secure edge computing-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 3774–3786, 2023.

[6] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6949–6960, Sep. 2022.

[7] S. Goudarzi, S. A. Soleymani, W. Wang, and P. Xiao, "UAV-enabled mobile edge computing for resource allocation using cooperative evolutionary computation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 5, pp. 5134–5147, Oct. 2023.

[8] 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, Nov. 2023.

[9] W. Fan, J. Liu, M. Hua, F. Wu, and Y. Liu, "Joint task offloading and resource allocation for multi-access edge computing assisted by parked and moving vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5314–5330, May 2022.

[10] F. Shirin Abkenar, P. Ramezani, S. Iranmanesh, S. Murali, D. Chulerttiyawong, X. Wan, A. Jamalipour, and R. Raad, "A survey on mobility of edge computing networks in IoT: State-of-the-art, architectures, and challenges," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 4, pp. 2329–2365, 4th Quart. 2022.

[11] Y. Zhang, J. Hu, and G. Min, "Digital twin-driven intelligent task offloading for collaborative mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3034–3045, Oct. 2023.

[12] L. Zhao, E. Zhang, S. Wan, A. Hawbani, A. Y. Al-Dubai, G. Min, and A. Y. Zomaya, "MESON: A mobility-aware dependent task offloading scheme for urban vehicular edge computing," *IEEE Trans. Mob. Comput.*, pp. 1–15, 2023.

[13] P. Li, Z. Xiao, X. Wang, K. Huang, Y. Huang, and H. Gao, "EPtask: Deep reinforcement learning based energy-efficient and priority-aware task scheduling for dynamic vehicular edge computing," *IEEE Trans. Intell. Veh.*, pp. 1–17, 2023.

[14] D. Van Huynh *et al.*, "URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7669–7682, Nov. 2022.

[15] Z. Yao, S. Xia, Y. Li, and G. Wu, "Cooperative task offloading and service caching for digital twin edge networks: A graph attention multi-agent reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3401–3413, Nov. 2023.

[16] X. Chen, G. Han, Y. Bi, Z. Yuan, M. K. Marina, Y. Liu, and H. Zhao, "Traffic prediction-assisted federated deep reinforcement learning for service migration in digital twins-enabled MEC networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3212–3229, Oct. 2023.

[17] V. Hassija, V. Chamola, V. Saxena, V. Chanana, P. Parashari, S. Mumtaz, and M. Guizani, "Present landscape of quantum computing," *IET Quantum Communication*, vol. 1, no. 2, pp. 42–48, Dec. 2020.

[18] J. A. Ansere, E. Gyamfi, V. Sharma, H. Shin, O. A. Dobre, and T. Q. Duong, "Quantum deep reinforcement learning for dynamic resource allocation in mobile edge computing-based IoT systems," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2023.

[19] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "RWP+: A new random waypoint model for high-speed mobility," *IEEE Commun. Lett.*, vol. 25, no. 11, pp. 3748–3752, Nov. 2021.

[20] N. Garg, A. K. Jagannatham, G. Sharma, and T. Ratnarajah, "Precoder feedback schemes for robust interference alignment with bounded CSI uncertainty," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 407–425, 2020.

[21] H. Zhang and L. Hanzo, "Federated learning assisted multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14 104–14 109, Nov. 2020.

[22] W. Wei, G. Cui, X. Yu, R. Liu, and X. Wang, "Asymmetric beampattern synthesis for rectangular planar array via window function design," *IEEE Trans. Signal Process.*, vol. 72, pp. 400–414, 2024.

[23] K. C. Okafor, B. Adebisi, A. O. Akande, and K. Anoh, "Agile gravitational search algorithm for cyber-physical path-loss modelling in 5G connected autonomous vehicular network," *Vehicular Communications*, vol. 45, p. 100685, Feb. 2024.

[24] 3rd Generation Partnership Project, "Study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, Sophia Antipolis, France, Tech. Rep., 2018.

[25] H. Zhao, A. Bazco-Nogueras, and P. Elia, "Coded caching gains at low SNR over nakagami fading channels," in *Proc. Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2021, pp. 26–32.

[26] S. Zhu, T. S. Ghazaany, S. M. R. Jones, R. A. Abd-Alhameed, J. M. Noras, T. Van Buren, J. Wilson, T. Suggett, and S. Marker, "Probability distribution of rician k-factor in urban, suburban and rural areas using real-world captured data," *IEEE Trans. Antennas Propag.*, vol. 62, no. 7, pp. 3835–3839, Jul. 2014.

[27] C. Zhong, X. Chen, Z. Zhang, and G. K. Karagiannidis, "Wireless-powered communications: Performance analysis and optimization," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 5178–5190, Dec. 2015.

[28] L. Zhou, H. Ma, Z. Yang, S. Zhou, and W. Zhang, "Unmanned aerial vehicle communications: Path-loss modeling and evaluation," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 121–128, Jun. 2020.

[29] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10 863–10 877, Oct. 2022.

[30] Z. Ning, Y. Yang, X. Wang, Q. Song, L. Guo, and A. Jamalipour, "Multi-agent deep reinforcement learning based UAV trajectory optimization for differentiated services," *IEEE Trans. Mob. Comput.*, pp. 1–17, 2023.

[31] S. K. Satpathy, V. Vibhu, B. K. Behera, S. Al-Kuwari, S. Mumtaz, and A. Farouk, "Analysis of quantum machine learning algorithms in noisy

channels for classification tasks in the IoT extreme environment," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 3840–3852, Feb. 2024.

[32] C. Park, W. J. Yun, J. P. Kim, T. K. Rodrigues, S. Park, S. Jung, and J. Kim, "Quantum multiagent actor–critic networks for cooperative mobile access in multi-UAV systems," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 20 033–20 048, Nov. 2023.

[33] S. Park, J. P. Kim, C. Park, S. Jung, and J. Kim, "Quantum multi-agent reinforcement learning for autonomous mobility cooperation," *IEEE Commun. Mag.*, pp. 1–7, 2023.