Variational Quantum Reinforcement Learning for Joint Resource Allocation of Blockchain-based Vehicular Edge Computing and Quantum Internet

Kening Zhang, Carman K. M. Lee, *Senior Member, IEEE*, Yung Po Tsang, *Member, IEEE*, Chun Ho Wu, *Member, IEEE*

***Abstract*—With the advances in artificial intelligence and communication technologies, vehicular edge computing (VEC), as a newly developed computing paradigm, is gaining more and more attention from both academia and industry. Complex demands and on-board applications need to be offloaded to edge servers for Quality of Experience (QoE). Nevertheless, the offloading process increases the risk of user privacy leakage, and the effectiveness of resource allocation algorithms is urgently desired in latency-sensitive tasks. To this end, we employ quantum key distribution (QKD) and blockchain to secure communication and computation, where key generation rate (KGR) associated with transmission and computation-aware is investigated for resource allocation problem. In consideration of the number of existing qubits and technical bottlenecks, we propose a tensor network preprocessing-based quantum deep reinforcement learning algorithm (TN-QDRL), which exploits amplitude encoding and the unique properties of quantum superposition and entanglement states to tackle the complex Markov decision process in a multi-dimensional state space. Additionally, we provide a search strategy for quantum state probabilistic transformations integrated with an improved Grover's algorithm. Simulation results indicate that our algorithm achieves a convergence speed that is 62.11% faster in high-dimensional real-world VEC scenarios and consumes 58.19% fewer quantum resources compared to other benchmarks.**

***Index Terms*—Vehicular edge computing (VEC), variational quantum circuits (VQC), quantum key distribution (QKD), Grover's algorithm, blockchain technology, quantum reinforcement learning.**

I. Introduction

With the burgeoning expansion of Internet-of-Vehicles (IoV), there has been a significant increase in applications that are computationally complex, delay-sensitive, and energy-sustainable, such as autonomous driving, smart parking (SP), and electric vehicle charging (EVC) [1]–[4]. The pervasive nature of these applications and services demands through vehicle-to-everything (V2X) technology to meet rigorous architectural expectations [5], [6], where the constrained computing availability in vehicles make management challenging. While traditional cloud computing frameworks can address bandwidth-heavy tasks, the long distance of the cloud servers from the vehicle makes them ineffective when latency-sensitive tasks arise, which poses a problem in improving the Quality of Experience (QoE) [7].

In response to these challenges, vehicular edge computing (VEC) offers a potent solution by enabling resource-constrained vehicles to offload computationally intensive tasks to edge servers (EDs) positioned at roadside units (RSUs) via wireless transmission links [8], [9]. This shift reduces latency and enhances vehicle responsiveness, improving the overall QoE and mitigating network congestion, critical for smart cities and intelligent transportation systems. However, as urban vehicle numbers increase, especially in areas requiring real-time data processing like parking lots and charging stations, the load on EDs escalates [10]. Deploying additional EDs is expensive, and it appears to be even more inefficient during off-peak times [11]. Moreover, efficient resource allocation, delay mitigation, and task management are essential in diverse vehicular environments [12]. These challenges can create bottlenecks, reduce efficiency, and increase costs within complex multidimensional state-action spaces. While deep reinforcement learning (DRL) addresses task offloading by modeling it as a Markov decision process (MDP) [13], its lengthy training times and rigid model parameters make it less suitable for dynamic VEC scenarios.

Furthermore, VEC reduces the distance of wireless transmission and increase the security of data to some extent, but it still lacks measures to guarantee the cybersecurity. In particular, with the development of quantum computers, such an up-and-coming technology can create a great threat to the traditional cryptography. For decades, cryptographic protocols such as Rivest–Shamir–Adleman (RSA) and Diffie-Hellman have been the cornerstone of secure communication over the internet [14]. These protocols leverage the difficulty of factoring large integers and computing discrete logarithms to establish secure connections and exchange keys between parties, respectively. It is regretted that quantum computers are able to perform many calculations simultaneously using the principles of quantum mechanics. Some existing algorithms like Shor's algorithm can efficiently factor large integers and solve discrete logarithm problems, which renders traditional key distribution schemes vulnerable to attacks [15]. Fortunately, quantum key distribution (QKD) [16] provides a viable

The authors would like to thank the Research and Innovation Office of the Hong Kong Polytechnic University for supporting the project (Project Code: RMGT). The work is also supported by Smart Traffic Fund of Transport Department of HKSAR Government (Project code: PSRI/67/2306/PR). *(Corresponding author: Yung Po Tsang).*

Kening Zhang, Carman K. M. Lee and Yung Po Tsang are with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: keningcs.zhang@connect.polyu.hk; ckm.lee@polyu.edu.hk; yungpo.tsang@polyu.edu.hk).

Chun Ho Wu is with the Department of Supply Chain and Information Management, The Hang Seng University of Hong Kong, Hong Kong (e-mail: jackwu@ieee.org).

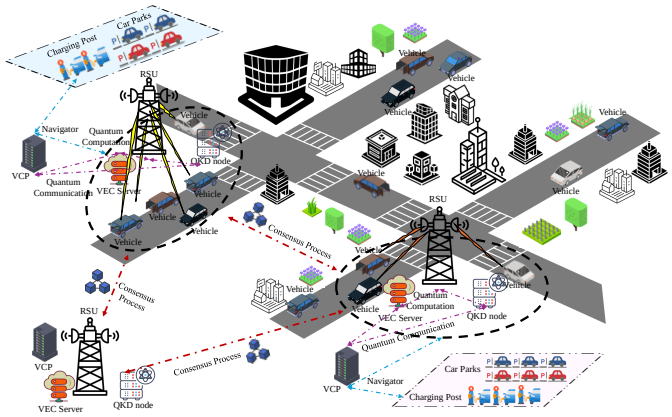

Fig. 1. Overview of the VEC scenario combining quantum communication and quantum computation for smart parking and charging. This figure illustrates a practical application scenario of the system architecture proposed later, in which VEC is integrated with quantum computing, quantum communication and blockchain technologies. QKD nodes (QNs) ensure secure communication between entities, providing quantum keys for encryption (QKD equipment layer). Blockchain-enabled consensus processes manage secure task allocation and record-keeping, ensuring data integrity and trustworthiness. Vehicles communicate with RSUs and VEC servers for task offloading and navigation, which is optimized by quantum algorithms (edge and blockchain layer). Charging posts and parking management systems are integrated to facilitate seamless charging and parking services for vehicles (vehicle layer).

solution to the paradigm of creating secure and reliable key distribution between EDs and vehicles by utilizing the Heisenberg's uncertainty principle [17] and the quantum no-cloning theorem [18]. To make full use of quantum technology for secure communications, the Quantum Internet (QI) is created [19], which consists of a combination of classic communication channels and quantum signalling transmissions to deliver long-term privacy preservation and meet the information-theoretic security (ITS) requirement for the transmission of sensitive data. Although existing research has focused on the RA in QKD [20]–[22], there are still some problems.

Firstly, while the RA problem of QKD in QI has been recognized, deploying secure quantum communication in VEC remains underexplored. Generating, storing, and distributing quantum keys (QKs) in dynamic scenarios, while optimizing transmission delay and computation assignment, is challenging. This requires synergizing QI and VEC to enhance system performance and efficiency. Although the quantum deep reinforcement learning (QDRL) algorithm holds promise for addressing this compute-intensive problem, its use in large-dimensional environments is limited by the few quantum bits available in current noisy intermediate-scale quantum (NISQ) devices [23]–[25]. Additionally, QKD ensures secure communication but does not protect offloading tasks in VEC, leaving data vulnerable to tampering or leaks. Ensuring data integrity in edge devices (EDs) and maintaining audit trails for data and QKs access are also crucial but overlooked in previous studies. The ideal scenario with technology is described in Fig. 1.

To address above-mentioned unresolved issues, We examine the joint RA problem to optimize key generation rate (KGR), energy consumption and transmission delay for QI and VEC-based system with blockchain technology. Blockchain technology [26] is an innovative-distributed ledger technology that enables secure and transparent recording of transactions and data across multiple computers. Its core features are decentralization, immutability and cryptographic security, which form the blockchain's infrastructure and operating mechanism together. With the support of blockchain, it is possible to address security issues at the computational level, protect the privacy and integrity of data, and improve the process of auditing the system. While there are studies on the RA problem for blockchain-based VEC [27], the threat of quantum computers has not been considered. For QI-enhanced VEC, we explore quantum features to develop a QK stake-based consensus mechanism to operate the blockchain system fairly. This method allows QKs to fully function, which saves the issue of needing to consume a lot of energy for mining in traditional blockchain systems. Inspired by quantum computing and tensor network (TN), a TN preprocessing-based quantum reinforcement learning (TN-QDRL) algorithm is designed to tackle this highly compute-complex problem.

In this paper, the main contributions of our paper are presented as follows:

1) We propose a novel blockchain-based VEC architecture with QI, named QIB-VEC. The designed system can support automatic adjustment of the KGR in QKD nodes, flexible offloading decision-making and adaptive computation allocation in EDs, which also ensures communication security, computational security, and data integrity, respectively, against being breached by quantum computers in the post-quantum era.
2) We develop a proof of QK stake (PoQKS) consensus mechanism, which utilizes the number of QKs held by nodes to safeguard the impartiality of the network. New records of task offloads, computational processes, and QK changes are uploaded to the blockchain and are regulated and audited.
3) We model a joint optimal RA problem in the QIB-VEC to maximize KGR and server resources, which is converted to an MDP under time-varying channel conditions. The designed TN-QDRL algorithm exploits the quantum parallelism to overcome high-dimensional difficulties and derive optimized resources in real time.
4) The extensive experiments show that TN-QDRL is clearly superior to the Q-learning algorithm and other benchmarks, the key generation rate can be maximized, and the latency is significantly reduced compared to other management schemes.

The reamaining structure of this article is as follows. Section II discusses related work and emphasize research gaps, which focuses on existing VEC and QI technologies. Section III describes the framework and system model of QIB-VEC. Section IV provides the detail of the TN-QDRL to address the RA problem. Section V devaluates the experimental results of the proposed system. Finally, conclusions and future work are discussed in Section VI.

II. Related Works

A. Vehicular Edge-Computing

VEC extends the capabilities and flexibility of cloud computing to the network edge and provides faster and more reliable services for mobile networks and vehicles, which has attracted a lot of discussions in both academia and industry. Since the resources of EDs are also limited, how to reasonably allocate the resources is a problem worth studying [28]. Li et al. [29] presented a particle swarm optimization-based method for VEC that minimizes delay and cost through effective resource scheduling and multi-objective optimization for autonomous driving. Mlika et al. [30] used network slicing (NS) and non-orthogonal multiple access (NOMA) to optimizing RA under various conditions of MEC-enabled vehicle networks. Feng et al. [31] suggested a reverse offloading framework that reduces system latency and enhances performance using optimized offloading strategies in the cooperative vehicle-infrastructure system. With the enhancement of these technologies, Quality-of-Service (QoS) is used as a metric to allocate the VEC's resources, which includes processing capability, latency, memory, bandwidth and energy [32]. Considering the security and efficiency challenges in VEC, Ju et al. [33] developed a secure offloading and resource allocation scheme using DRL to optimize resource use and enhance data security. Nan et al. [34] proposed two algorithms, A-TARFD and L-TARFD, to optimize offloading decisions and resource distribution, demonstrating near-optimal performance across various scenarios. Yang et al. [35] designed a parallel intelligence-driven resource scheduling scheme that enhances computation efficiency using an adaptive particle swarm with a genetic algorithm based on the dual dependencies of timing and data.

B. Quantum Internet

QI [19] contains a combination of classical channels and quantum secure channels, which is the main technology for secure transmission of private information among in the post-quantum era. As a mature technology in the QI, QKD is already available for industrial and commercial applications [20]. In terms of QKD deployment, a satellite-based QKD network has completely undergone experimental confirmation [36]. There are also multiple fiber-based connection like Beijing-Shanghai [37], SwissQuantum [38], SECOQC [39] and Tokyo [40] QKD networks that have even been effectively implemented. In the management of QKD networks, software-defined networking (SDN) can provide a more functional approach to programmable management. Aguado et al. [41] exploits the SDN technique and proposes a new scheme based on virtual network function and time-shared method for the QKD RA problem. Cao et al. [42] designed a three-stage heuristic algorithm, which integrates quantum the key pool to distribute sufficient secret keys efficiently in resource-constrained wavelength-division multiplexing network. For QKD network operation, QKD-as-a-Service (QaaS) [43] is suggested to enhance cybersecurity and achieve greater cost productivity in SDN control, IoT, virtual network and so on. To mitigate the discrepancy between the limited key production capacity and the variable needs for keys in quantum secure communication, the implementation of high-performing distribution strategies for QKD resources is deemed essential. In [44], a reinforcement learning method based on QK pooling threshold is presented to manage key resources and dynamic key services, which effectively reduces the rate of burst business congestion. Cao et al. [21] replaces the classic heuristic algorithm with a novel reinforcement learning algorithm for multi-tenant provisioning-based problems in QKD networks when multiple tenant requests are delivered in real time. Xu et al. [20] uses federated learning to manage stochastic resources in QKD's managers and controllers, optimizing deployment costs and enhancing communication security in QI.

C. Research Gap

In VEC, the focus is on optimizing resource allocation and improving technologies to reduce latency, enhance performance, and ensure data security. For QI, we discuss the application and management of QKD, highlighting strategies to optimize resource allocation and safeguard network security using technologies like Software-Defined Networking (SDN) and reinforcement learning. However, there are still some research gaps.

First, studies on VEC does not take into account the important factor of post-quantum security, which can prove insufficient when confronted with the threats posed by quantum computing. This oversight implies that the security assessments and recommendations provided may not retain their effectiveness and reliability in a future quantum computing environment.

Second, although current models and algorithms of the RA problem in QKD exhibit certain superiority and innovation, the study does not sufficiently account for the complexity and diversity of real-world application scenarios. For instance, factors such as network topology, transmission distance, environmental noise, and equipment stability in actual scenarios can significantly impact the performance and resource requirements of QKD. Ignoring these practical conditions may result in resource allocation strategies that fail to achieve the expected outcomes in real-world applications, which limits the adoption and implementation of QKD technology in practical settings. Naturally, VEC offers an ideal bridge for integrating QKD resource allocation with real-world scenarios.

Third, despite the fact that QKD technology ensures communication security, it does not provide integrity assurance for data storage and processing. Existing research primarily focuses on the application of QKD in communication security, while the data traceability requirements in the context of VEC have been largely overlooked, which is crucial for identifying attackers. Ensuring the integrity and traceability of data storage and processing in VEC scenarios is particularly critical. Therefore, studying the data traceability requirements when combining QKD with VEC can significantly enhance the security and reliability of data in distributed computing environments.

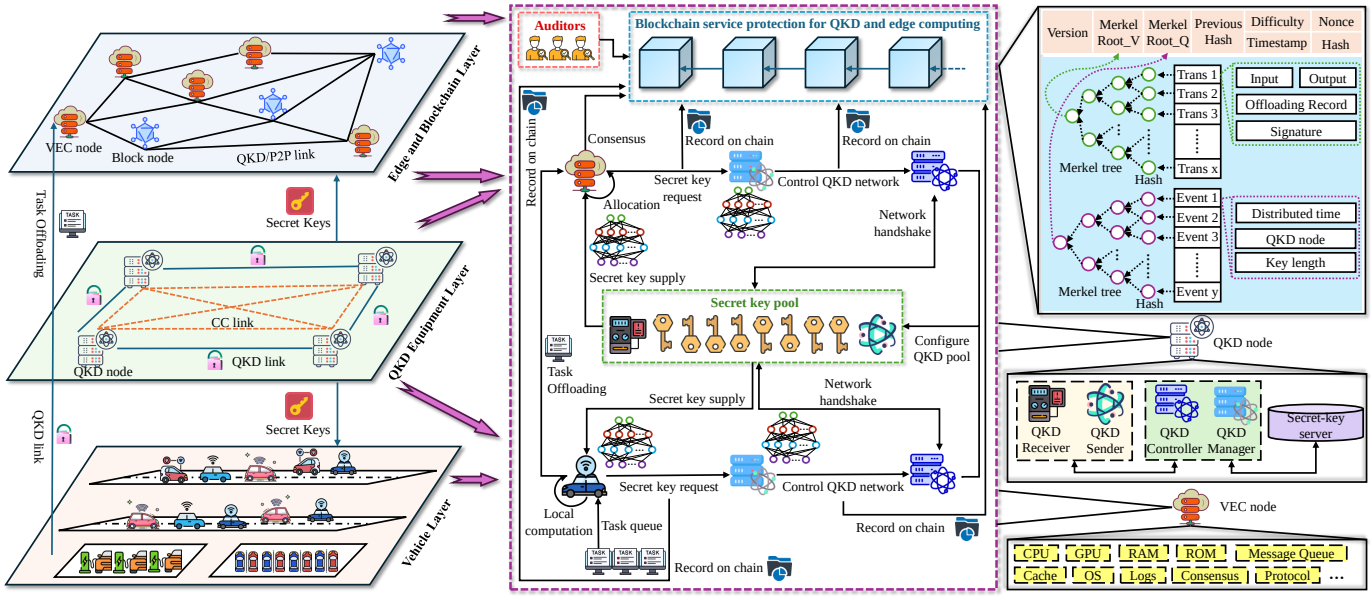

Fig. 2. Framework of the QIB-VEC

III. System Model

In this section, we present the total system architecture of QIB-VEC. The system presented in this work is based on a combination of well-established foundational concepts from existing research and novel adaptations designed to address the unique challenges and requirements of the current issues. In the model, the QKD network module, the VEC module, the trust blockchain module and the corresponding RA problem are defied by the formulas in detail.

A. System Scenario

As shown in Fig. 2, we consider a three-layer blockchain-based MEC system with QI, comprising the vehicles layer (VL), the QKD equipment layer (QL), and the edge and blockchain layer (EBL). In this setup, QKD and blockchain secure network communications and computation offloading while ensuring system auditability. The QKD network dynamically allocates resources based on real-time feedback from EBL and IL to improve the key generation rate (KGR). During this process, quantum signals are produced by exploiting quantum optical processes and distributed through optical fibers. In the EBL, an edge device (ED), a QKD equipment, and a vehicle crowdsourcing platform (VCP) are deployed around each RSU. EDs serve as full blockchain nodes, meeting the computation and mining requirements due to their ample resources. Quantum simulators in EDs execute quantum algorithms to accelerate neural network convergence. The VCP manages nearby car parks and charging posts, offering city-wide parking and charging recommendations. Records of key generation, storage, and update processes in the QKD network, along with offloaded computational tasks from vehicles in the VEC network, are treated as 'transactions' in the blockchain system. Once these transactions are packed into a block, they are verified by other full nodes and uploaded to the blockchain using the proposed PoQKS consensus mechanism. In order to improve the clarity of the mathematical derivation and its impact on the problem at hand, we have elaborated the variables and their interpretations in Table I.

B. QKD Network Module

In the QKD network, the type of the trusted repeater QKD network we adopt is a backbone or metro network, both of which are discussed in [37]. Here, this paper adopts the existing quantum communication technology as a premise, assuming that the generation, modulation and distribution of quantum signals can be realized through a mature QKD system. Assume that there are a total of V vehicles, which can be denoted as $\mathbb{V} = \{v_1, \ldots, v_x, \ldots, v_V\}$. Suppose there are Q QNs in the QI, which are represented by $\mathbb{Q} = \{q_1, \ldots, q_y, \ldots, q_Q\}$. The EDs are considered to have a total of E, which can be expressed as $\mathbb{E} = \{e_1, \ldots, e_z, \ldots, e_E\}$. Without loss of generality, each variable is hypothesised to be a representation over a time interval $[\varepsilon, \varepsilon+1]$, $\varepsilon \in \{0, 1, 2, \ldots\}$. Let $l_{i,j}(\varepsilon)$ denote the channel length (km) from the QN i to the target device j at time slot ε, where $i \in \mathbb{Q}, j \in \mathbb{V} \cup \mathbb{E}$. Based on quantum optics principles and the Beer-Lambert law [45], the entanglement pair generation rate from the QN i to the receiver j at time slot ε can be given by

$$\kappa_{i,j}^{ent}(\varepsilon) = v_s \cdot \sin^2(\vartheta^k(\varepsilon)) \cdot e^{-\sigma \cdot l_{i,j}(\varepsilon)} \tag{1}$$

where v_s is the maximum output rate of the quantum source, $\vartheta^k(\varepsilon)$ means the control parameter for quantum state preparation at time slot (ε), and σ denotes the source-specific attenuation coefficient. Furthermore, based on the principle of wave interference, the clarity and contrast of the interference pattern formed due to coherent superposition of quantum states is called interference visibility, which can be expressed as

$$\varpi_I(\varepsilon) = \frac{|\cos(\vartheta^I(\varepsilon))|}{1+\rho} \tag{2}$$

TABLE I
VARIABLES AND THEIR EXPLANATIONS

Variable	Description
V	Total number of vehicles in the system ($\mathbb{V} = \{v_1, \ldots, v_x, \ldots, v_V\}$).
Q	Total number of QKD nodes ($\mathbb{Q} = \{q_1, \ldots, q_y, \ldots, q_Q\}$).
E	Total number of edge devices ($\mathbb{E} = \{e_1, \ldots, e_z, \ldots, e_E\}$).
ε	Time interval ($[\varepsilon, \varepsilon + 1]$).
$l_{i,j}(\varepsilon)$	Channel length (km) from QKD node i to device j at time ε.
$\kappa_{i,j}^{ent}(\varepsilon)$	Entanglement pair generation rate from QKD node i to receiver j at time ε.
v_s	Maximum output rate of the quantum source.
$\vartheta^k(\varepsilon)$	Control parameter for quantum state preparation at time ε.
σ	Source-specific attenuation coefficient.
$\varpi_I(\varepsilon)$	Interference visibility, measuring clarity and contrast of quantum interference patterns.
$\eta_i(\varepsilon)$	Detector efficiency of QKD node i, reflecting detection success probability.
$\eta_{sys}(\varepsilon)$	System efficiency, combining all detector efficiencies and other operational components.
$p_{i,j}^{ts}(\varepsilon)$	Probability of successful quantum state transmission from QKD node i to receiver j at time ε.
$\kappa_{i,j}^{qbe}(\varepsilon)$	QBER between QKD node i and receiver j at time ε.
$s_{i,j}^{q}(\varepsilon)$	Signal strength, rate of correctly received qubits contributing to key generation.
$f_{i,j}^{n}(\varepsilon)$	Noise function in the quantum channel from QKD node i to receiver j, depending on angular velocity ω and other factors.
$N_{i,j}(\varepsilon)$	Total noise in the channel from QKD node i to receiver j, including baseline noise N_0.
$h_{i,j}^{e}(\varepsilon)$	Binary entropy, representing the uncertainty of a bit given the QBER $\kappa_{i,j}^{qbe}(\varepsilon)$.
$\triangle_{i,j}^{sec}(\varepsilon)$	Security margin in the QKD system, accounting for signal filtering and statistical fluctuations.
$\kappa_{i,j}(\varepsilon)$	KGR from QKD node i to receiver j considering system efficiency and security margin.
$\gamma_x^{v,m}(\varepsilon)$	SINR of vehicle v_x to RSU m.
$\nu_x^{v,m}(\varepsilon)$	Transmission rate of data from vehicle v_x to RSU m.
$t_x^{c,n}(\varepsilon)$	Time required for local computation of a task by vehicle v_x.
$t_{x,m}^{tcb,n}(\varepsilon)$	Total time for task transmission, computation, and backhaul processing.
$T_{x,m}^{sum}(\varepsilon)$	Total latency cost for a task handled by vehicle v_x via RSU m, including transmission, computation, and backhaul latencies.

where $\vartheta^I(\varepsilon)$ is the phase difference of the quantum state at time slot ε and ρ is phase noise parameter.

The detector efficiency in QN i, as the probability that the detector successfully detects an incoming quantum state, can be presented as

$$\eta_i(\varepsilon) = \eta_{int} \cdot (1 - \mu^{tem} F(\varepsilon)) \cdot (F(\varepsilon) - F_0) \tag{3}$$

where η_{int} is the detection efficiency at the reference temperature F_0, μ^{tem} is the temperature coefficient, and $F(\varepsilon)$ is the operating temperature at the time slot ε. Therefore, the total efficiency of the system is

$$\eta_{sys}(\varepsilon) = \eta_{ot} + \sum_{i \in \mathbb{Q}} \eta_i(\varepsilon) \tag{4}$$

where η_{ot} is the system efficiency of the whole system except detectors in QNs.

Based on the Beer-Lambert law used in classical fiber-optic communications, the probability of successful transmission of a quantum state from the QN i to the receiver j in a quantum channel at time slot ε is given by

$$p_{i,j}^{ts}(\varepsilon) = e^{-\frac{\alpha(\lambda) \cdot l_{i,j}(\varepsilon)}{10}} \tag{5}$$

where λ is the wavelength, $\alpha(\lambda)$ is the wavelength-dependent attenuation coefficient (dB/km).

The signal strength in a QKD system typically refers to the rate of correctly received quantum bits (qubits) that contribute to key generation, which can be calculated by the quantum source's output rate and the transmission and detection efficiencies from the QN i to the receiver j. The formula is denoted as

$$s_{i,j}^{q}(\varepsilon) = \kappa_{i,j}^{ent}(\varepsilon) \cdot p_{i,j}^{ts}(\varepsilon) \cdot \eta_{sys}(\varepsilon) \tag{6}$$

The noise function $f^n(\cdot)$ in the field of quantum communication depends on the specific physical environment and type of channel. Based on quantum communication principles, we give the following formula that varies with angular velocity ω at time slot (ε) to present interactions of photons and how noise affects quantum channels from QN i to receiver j.

$$f_{i,j}^{n}(\varepsilon) = A_{i,j}(\omega) e^{-\beta_{i,j}(\omega\varepsilon)} + B_{i,j}(\omega) \cos(\mu^q(\omega\varepsilon) + \varphi) \tag{7}$$

where $A_{i,j}(\omega)$ describes the noise amplitude related to frequency, which is associated with scattering and absorption of the optical signal. $\beta_{i,j}(\omega)$ is the decay coefficient, $B_{i,j}(\omega)$ indicates the amplitude of oscillation, $\mu^q(\omega\varepsilon)$ is the angular frequency, and φ is the phase offset.

Then, the channel noise from QN i to receiver j can be represented with the baseline noise N_0 as follows

$$N_{i,j}(\varepsilon) = N_0 + \int f_{i,j}^{n}(\omega, \varepsilon) \mathrm{d}\varepsilon \tag{8}$$

Accordingly, the quantum bit error rate (QBER) formula is rooted in Shannon's Information Theory, particularly in how error probabilities affect communication systems. It is expressed as

$$\kappa_{i,j}^{qbe}(\varepsilon) = \frac{1}{2}(1 - \varpi_I(\varepsilon) \cdot e^{\eta_{sys}(\varepsilon) \cdot p_{i,j}^{ts}(\varepsilon) \cdot \frac{s_{i,j}^{q}(\varepsilon)}{N_{i,j}(\varepsilon)}}) + \epsilon^{err} \tag{9}$$

where ϵ^{err} includes other sources of errors.

Here, we use a binary entropy to represent the uncertainty of each bit at a given $\kappa_{i,j}^{qbe}(\varepsilon)$, which is denoted as

$$\begin{aligned} h_{i,j}^{e}(\varepsilon) = &- \kappa_{i,j}^{qbe}(\varepsilon) \log_2 \kappa_{i,j}^{qbe}(\varepsilon) \\ &- (1 - \kappa_{i,j}^{qbe}(\varepsilon)) \log_2 (1 - \kappa_{i,j}^{qbe}(\varepsilon)) \end{aligned} \tag{10}$$

According to the security metric proposed by [46], we utilize the $\epsilon-$security to estimate the security of the QKD process in QIB-VEC based on the trace distance. Thereby, the security margin $\triangle^{sec}(\varepsilon)$ in the QKD system can be expressed as

$$\triangle_{i,j}^{sec}(\varepsilon) = \sqrt{\frac{\mu_1}{n_{sec}}} + \mu_2 \cdot h_{i,j}^{e}(\varepsilon) + \epsilon_{sec} \tag{11}$$

where n_{sec} is the total number of signals or the length of the filtered key. ϵ_{sec} is a very small value indicating the security parameter used to ensure the $\epsilon_{sec}-$security. μ_1 and μ_2 are correlation coefficients for statistical fluctuations.

Considering the security margin mentioned in (11), the KGR from the QN i to the receiver j with the signal detection efficiency η_i of the system can be expressed as

$$\kappa_{i,j}(\varepsilon) = \eta_i(\varepsilon) \cdot \kappa_{i,j}^{ent}(\varepsilon) \cdot (1 - \kappa_{i,j}^{qbe}(\varepsilon)) - \triangle_{i,j}^{sec}(\varepsilon) \quad (12)$$

C. VEC Module

For vehicles in the city, NOMA technology [7] is typically considered as the communication protocol between the vehicle and the RSU. When different vehicles need to offload computing tasks to the same RSU, communications will interfere with each other and contend for the bandwidth. In our model, each RSU is equipped with an ED, thus the set of RSUs can be represented as $\mathcal{M} = \{1, \ldots, m, \ldots, M_r\}$, where $|\mathcal{M}| = E$. Assume that signal-to-interference-plus-noise ratio (SINR) from the vehicle v_x to the RSU m can be represented as

$$\gamma_x^{v,m}(\varepsilon) = \frac{p_x^{v,m}(\varepsilon) g_x^{v,m}(\varepsilon)}{\sum_{i \neq x, i \in \mathbb{N}^*, i \leq V} p_i^{v,m}(\varepsilon) g_i^{v,m}(\varepsilon) + \sigma_v^2} \quad (13)$$

where $p_x^{v,m} \in [p_{min}^v(\varepsilon), p_{max}^v(\varepsilon)]$ and $g_x^{v,m} \in [g_{min}^v(\varepsilon), g_{max}^v(\varepsilon)]$ mean the transmission power and channel gain, respectively. σ_v represents the additive white Gaussian noise (AWGN). $\sum_{i \neq x, i \in \mathbb{N}^*, i \leq V} p_i^{v,m}(\varepsilon) g_i^{v,m}(\varepsilon)$ implies mutual interference between vehicle v_x and other vehicles connected to the RSU m. Then, the transmission rate based on the Shanon–Hartley formula from the v_x to the RSU m can be denoted as

$$\nu_x^{v,m}(\varepsilon) = B_x^{v,m} \log_2 \left(1 + \gamma_x^{v,m}(\varepsilon)\right) \quad (14)$$

where $B_x^{v,m}$ indicates the bandwidth allocated to the vehicle v_x from the RSU m.

When the vehicle needs a computing task, it has two options: local computing and offloading computing. Accordingly, we divide the delay of the whole system into three parts: transmission latency (TL), computation latency (CL) and backhaul latency (BL). Consider that the local CPU of the vehicle v_x has a computing capability of $c_x^v(\varepsilon)$ (in CPU cycles/s), and that the task that needs to be computed requires a resource consumption of $f_n(\varepsilon)$. Then, the time for local computation is expressed as

$$t_x^{c,n}(\varepsilon) = \frac{f_n(\varepsilon)}{c_x^v(\varepsilon)} \quad (15)$$

If a vehicle needs to offload a task to an ED, it needs to transmit the task first. Assume the task size of that requires being computed from the vehicle v_x to the ED $e_z(\varepsilon)$ is $s_n^v(\varepsilon)$. Then, the total time to process this task is the sum of TL, CL and BL, which can be represented as

$$t_{x,m}^{tcb,n}(\varepsilon) = \underbrace{\frac{s_{x,z}^v(\varepsilon)}{\nu_x^{v,m}(\varepsilon)}}_{\text{TL}} + \underbrace{\frac{f_n(\varepsilon)}{c_z^e(\varepsilon)}}_{\text{CL}} + \underbrace{\max\left\{\frac{s_{z,x}^e(\varepsilon)}{\nu_x^{v,m}(\varepsilon)}, \varsigma \cdot s_{z,x}^e(\varepsilon)\right\}}_{\text{BL}} \quad (16)$$

where $s_{x,z}^v(\varepsilon)$ and $s_{z,x}^e(\varepsilon)$ are the task size to be offloaded and the result size of the completed computation at time slot ε, respectively. $c_z^e(\varepsilon)$ (in CPU cycles/s) means the computation capability of the ED $e_z(\varepsilon)$. ς (ms/bit) denotes the transmission delay of the unit data size from the VCP to the vehicle. When the data reaches the VCP or the vehicle, we take the longer time as the delay in completing the task.

Therefore, the total time cost in the VEC module can be calculated by

$$T_{x,m}^{sum}(\varepsilon) = I_{x,m}(\varepsilon) t_x^{c,n}(\varepsilon) + (1 - I_{x,m}(\varepsilon)) t_{x,m}^{tcb,n}(\varepsilon) \quad (17)$$

where the $\mathbf{I^x}(\varepsilon) = (I_{x,m})$ means the offloading strategy of the vehicle v_x at the RSU m. Accordingly, the total time cost needs to satisfy the constraint of not exceeding the maximum delay value.

Algorithm 1: Proof of QK Stake (PoQKS) Consensus Mechanism

Setup each node i gets initial QK number $qk(i)$.
Setup initial weight w_i of node i.
Setup total system key volume $T_{qk} \leftarrow qk.get\_sum()$.
Setup the set of nodes in the production I_p and validation I_v.
for each node i in I_p **do**
  Calculate the quantum stake weight $w_i \leftarrow \frac{qk(i)}{T_{qk}}$.
  Get the number of votes v_i of per node.
  Calculate the vote with the weight $v^q(i) \leftarrow w_i \cdot v_i$.
end
Reach consensus after more than 2/3 of the voting weight agreed.
Get the node p that is qualified for production $p = v^q.get\_nouce(v^q.get\_max())$.
Calculate the rewards received by the nodes $qr_p \leftarrow \alpha_1 \cdot w_p$.
Sort the nodes in I_p by the quantum stake $I_p \leftarrow I_p.sort\_bts()$.
for each node j in I_p **do**
  if $isValid(p.get\_block()) != True$ **then**
    Reduce proportionally the quantum stake $qk(p) \leftarrow qk(p) - qr_p - \beta \cdot w_p$.
    Update the quantum stake $qk(j) \leftarrow qk(j) + \alpha_2 \cdot w_p$.
    break.
    else
      Update the quantum stake $qk(j) \leftarrow qk(j) + \alpha_3 \cdot w_p$.
    end
  end
end
Distribute the reward to production node $qk(p) \leftarrow qk(p) + qr_p$.
Update total system key volume $T_{qk} \leftarrow qk.get\_sum()$.

D. Trust Blockchain Module

By offloading parking and charging tasks to the ED, vehicles can have better computing experiences. Although quantum communication ensures secure communication, the process of computing in the ED also carries the risk of privacy leakage. The integration of blockchain in QIB-VEC can improve

security and privacy protection. The ED is selected based on the number of QKs to prevent malicious data misuse, since it has enough resources to act as a blockchain node. In addition, transaction records within the system are handled by the blockchain, including generating blocks and validating transactions through a consensus process.

During block generation phase, suppose the wth ED is used as a node to produce the blockchain. Denote the CPU-cycle frequency assigned to produce blocks and processing density as c_m^b (in cycles/s) and ρ^b (in cycles/bit). Then, the production time for one block is presented as

$$t_b^g(\varepsilon) = \frac{\rho^b(\varepsilon)s^b}{c_m^b(\varepsilon)} \tag{18}$$

where s^b is one block size that contains block header, transactions, events and so on as shown in Fig. 2.

In the consensus process, we propose a PoQKS algorithm based on QK stakes to reach consensus, which means the more QKs a node has, the more it communicates with vehicles and therefore should have a greater decision weight. In this consensus, the node obtains more QK stakes by completing offloading tasks, uses these stakes to perform weight calculations, and then participates in the consensus decision-making of the network, as detailed in Algorithm 1. Assume that there are $|\mathcal{W}|$ nodes competing for votes and the number of QKs possessed by a node w is $n_w^q(\varepsilon)$. Then, the propagation time is given by

$$t_b^c(\varepsilon) = n_w^q(\varepsilon) \cdot c^q(\varepsilon) + \max_{i \neq w, i \in \mathcal{W}} \frac{s^v(\varepsilon)}{B_{w,i}(\varepsilon)} \tag{19}$$

where $\mathcal{W} \subseteq \mathbb{E}$, $c^q(\varepsilon)$ is the delay required to compute the weight of a unit key at the time slot ε, and $s^v(\varepsilon)$ means the data size of voting information at the time slot ε. $B_{w,i}(\varepsilon)$ represents the bandwidth between the node w and i.

For block verification, we concentrate only on the cryptographic operation cost. We assume that $n_v(\varepsilon)$ and s_b^h are the number of transactions to be validated and the size of the block header, respectively. Then, the verification time is given by

$$t_b^v(\varepsilon) = (\frac{s_b^h}{\triangle s_v} + n_v(\varepsilon)) \triangle t_v \tag{20}$$

where $\triangle s_v$ and $\triangle t_v$ denote the size and the validation time of one single transaction. Here, we adopt delay/time to finality (DTF) to measure the delay of this module, which is expressed as

$$T_w^b(\varepsilon) = (t_b^g(\varepsilon) + t_b^c(\varepsilon) + t_b^v(\varepsilon))I_w^b(\varepsilon) \tag{21}$$

In equation (21), $\mathbf{I^b}(\varepsilon) = (I_w^b(\varepsilon))$, where $I_w^b(\varepsilon)$ is an indicator variable to signify whether the node w is elected as a production node or not.

E. Problem Formulation

How to accomplish RA of QKD and VEC is an understudied issue. Here, we aim to jointly optimize quantum resources, VEC resources and task offloading to maximize KGR and minimize latency of offloading and blockchain. Specifically, physical environment and type of quantum channel, communication and computation resources, and offloading strategies should be considered. Mathematically, we define the following formula to optimize the objective value of QIB-VEC.

$$\mathfrak{F} = \frac{1-\xi_1}{\sum_{i\in\mathbb{Q}}\sum_{j\in\mathbb{V}\cup\mathbb{E}}\xi_2\kappa_{i,j}} + \xi_1(\sum_{x\in\mathbb{V}}\sum_{m\in\mathcal{M}} T_{x,m}^{sum} + \sum_{w\in\mathcal{W}} T_w^b) \tag{22}$$

where $\xi_1(0 < \xi_1 < 1)$ means a weighting factor used to co-optimize the KGR and latency together, and ξ_2 is a meshing factor that map the different unit scales to the same level. Since we need to minimize the objective function here, the inverse of KGR is chosen as part of the objective when optimizing the QKD resource. Specifically, the joint optimization problem of signal strength $\mathbf{s^q} = (s_{i,j}^q)$, ED offloading selection $\mathbf{I^x}$, task rate allocation $\mathbf{v} = (\nu_x^{v,m})$, CPU-cycle frequency $\mathbf{c} = (c_z^e)$, block miner selection $\mathbf{I^b}$ is presented by

$$\begin{aligned}
&\min_{\mathbf{s^q},\mathbf{I^x},\mathbf{v},\mathbf{c},\mathbf{I^b}} \ \mathfrak{F} \\
&s.t.\ C1: p_{i,j}^{ts}(\varepsilon) \in [0,1], \forall i \in \mathbb{Q}, \forall j \in \mathbb{V}\cup\mathbb{E}, \\
&\quad C2: \kappa_{i,j}(\varepsilon) \geq 0, \forall i \in \mathbb{Q}, \forall j \in \mathbb{V}\cup\mathbb{E}, \\
&\quad C3: p_i^{v,m}(\varepsilon) \leq p_{max}^v, \forall i \in \mathbb{V}, \forall m \in \mathcal{M}, \\
&\quad C4: g_i^{v,m}(\varepsilon) \leq g_{max}^v, \forall i \in \mathbb{V}, \forall m \in \mathcal{M}, \\
&\quad C5: T_{x,m}^{sum}(\varepsilon) \leq T_{x,max}(\varepsilon), \forall m \in \mathcal{M}, \\
&\quad C6: T_w^b(\varepsilon) \leq T_{max}^b(\varepsilon), \forall w \in \mathcal{W}, \\
&\quad C7: \sum_{x\in\mathbb{V}} B_x^{v,m}(\varepsilon) = B^{v,m}(\varepsilon), \\
&\quad C8: \sum_{i\in\mathbb{W}} B_{w,i}(\varepsilon) \leq B_{Max}^w(\varepsilon) - B^{v,w}(\varepsilon), \\
&\quad C9: I_{x,m}(\varepsilon) \in \{0,1\}, \forall m \in \mathcal{M}, \\
&\quad C10: \sum_{m\in\mathcal{M}} I_{x,m}(\varepsilon) = 1, \\
&\quad C11: I_w^b(\varepsilon) \in \{0,1\}, \forall w \in \mathcal{W}, \\
&\quad C12: \sum_{w\in\mathcal{W}} I_w^b(\varepsilon) = 1, \\
&\quad C13: \nu_x^{v,m}(\varepsilon) \geq 0, \forall x \in \mathbb{V}, \forall m \in \mathcal{M}, \\
&\quad C14: c_z^e(\varepsilon) \geq 0, c_m^b(\varepsilon) \geq 0, c^q(\varepsilon) \geq 0, \\
&\quad C15: s^b(\varepsilon) \geq 0, s_b^h(\varepsilon) \geq 0, s^v(\varepsilon) \geq 0, \\
&\quad C15: n_w^q(\varepsilon), n_v(\varepsilon) \in \mathbb{N}^*
\end{aligned} \tag{23}$$

In the equation (23), constraint C1 and C2 ensure that keys in the QKD module make practical sense. C3 and C4 ensures that the energy and channel gains are within a certain range of values. C5 and C6 represents the maximum latency tolerated by the vehicles in the VEC and blockchain modules. C7 indicates that ED's bandwidth is allocated to connected vehicles, while C8 ensures that the network is not overloaded when blockchain nodes reach consensus. C9, C10, C11, and C12 represent the offloading strategy of vehicles and the selection strategy of the block miner. These constraints ensure that each vehicle can only offload tasks to one ED at a time, and that each block can only be generated by one node at the same moment. However, these variables make this problem non-convex, causing great difficulty in solving it. Each additional vehicle or node represents an exponential increase in computational complexity, which produces the

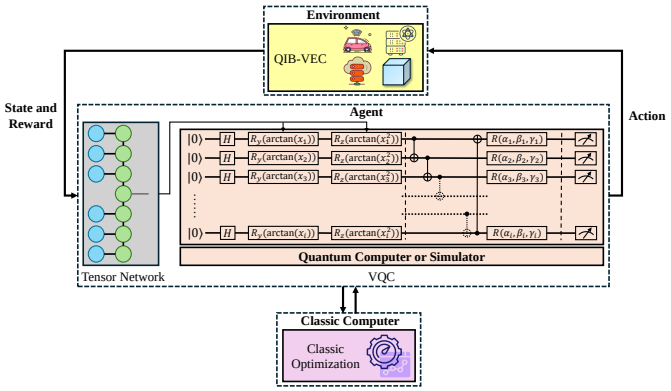

Fig. 3. Framework of the proposed TN-QDRL.

curse of dimensionality. Thus, this is a NP-hard problem. To make the solution more efficient, we design a TN-QDRL algorithm to solve this practical dynamic problem.

IV. Proposed TN-QDRL Algorithm Design for RA in QIB-VEC

Traditional algorithms often encounter challenges such as high computational costs, local optima, high-dimensional spaces, and non-convexity when dealing with (23). Particularly, for dynamic and real-time network states and data, traditional DRL methods still need to train large-scale parameters. By transforming the problem as an MDP, we can leverage QDRL to search for optimal strategies effectively. variational quantum circuit (QVC) is utilized to represent strategies and value functions, which shows unique advantages in solving complex MDP issues.

A. Overview of Reinforcement Learning

To mitigate the complexity posed by sophisticated VEC environments, the above problem is reformulated using the MDP model. Subsequently, the QDRL approach is employed to accelerate the convergence of the system rewards by harnessing the power of quantum mechanics and the flexibility.

Based on the trial-and-error methodology, the agent in the DRL algorithm can learn a lot about how the complex system is interconnected with the environment to make optimal decisions. After mathematical modelling, the state of the system can be represented and transferred to the next state after agent's action. By constantly interacting with the environment during the training process, the DRL agent will continue to achieve higher returns and eventually converge. The whole historical process of MDP is a sequence $h^{(i)} = (\mathfrak{s}_0^{(i)}, \mathfrak{a}_0^{(i)}, \mathfrak{r}_0^{(i)}, \mathfrak{s}_1^{(i)}, \mathfrak{a}_1^{(i)}, \mathfrak{r}_1^{(i)}, \ldots, \mathfrak{s}_n^{(i)}, \mathfrak{a}_n^{(i)}, \mathfrak{r}_n^{(i)})$. An MDP process can consist of a quintuple $< \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R}, \varrho >$, where $\mathbb{S}$ means the system state set, $\mathbb{A}$ is the action set of the agent, $\mathbb{P}(\mathfrak{s}'|\mathfrak{s}, \mathfrak{a})$ denotes the transaction probability set of moving from one state $\mathfrak{s}$ at time ε to another $\mathfrak{s}'$ at time $\varepsilon+1$ after an action $\mathfrak{a}$ has been taken, $\mathbb{R}(\mathfrak{s}, \mathfrak{a})$ represents the reward function associated with taking an action $\mathfrak{a}$ in a given state $\mathfrak{s}$, and $\varrho \in (0,1)$ is the discount factor. Specifically, each critical component of the MDP can be expressed in detail as follows.

1) State Space: At each discrete time ε, the agent can get a current state $\mathfrak{s}(\varepsilon) \in \mathbb{S}$ by observing the environment of QIB-VEC, where the state consists of a tuple with six components, including the entropy $\mathcal{H}(\varepsilon)$, transmission power $\mathcal{P}(\varepsilon)$, channel gain $\mathcal{G}(\varepsilon)$, security margin $\mathcal{F}(\varepsilon)$, local computation capability $\mathcal{L}(\varepsilon)$ and edge computation capability $\mathcal{E}(\varepsilon)$. Specifically, these components represent that $\mathcal{H}(\varepsilon) = \{h_{i,j}^e(\varepsilon), i = 1,2,\ldots,Q, j = 1,2,\ldots,V+E\}$, $\mathcal{P}(\varepsilon) = \{p_x^{v,m}(\varepsilon), x = 1,2,\ldots,V, m = 1,2,\ldots,M\}$, $\mathcal{G}(\varepsilon) = \{g_x^{v,m}(\varepsilon), x = 1,2,\ldots,V, m = 1,2,\ldots,M\}$, $\mathcal{F}(\varepsilon) = \{\triangle_{i,j}^{sec}, i = 1,2,\ldots,Q, j = 1,2,\ldots,V+E\}$, $\mathcal{L}(\varepsilon) = \{c_x^v(\varepsilon), x = 1,2,\ldots,V\}$, and $\mathcal{E}(\varepsilon) = \{c_z^e(\varepsilon), z = 1,2,\ldots,E\}$. Then, the state $\mathfrak{s}(\varepsilon)$ is defined as

$$\mathfrak{s}(\varepsilon) \triangleq \{\mathcal{H}(\varepsilon), \mathcal{P}(\varepsilon), \mathcal{G}(\varepsilon), \mathcal{F}(\varepsilon), \mathcal{L}(\varepsilon), \mathcal{E}(\varepsilon)\} \tag{24}$$

2) Action Space: The action space in this MDP for QIB-VEC typically includes all possible actions that can be taken at any given state. These actions are decisions made by the system to optimize performance and RA. The agent performs an action $\mathfrak{a}(\varepsilon) \in \mathbb{A}$ based on an optimal strategy π after observing the environment, which is composed of four dimensions: detector efficiency allocation $\mathcal{A}^q(\varepsilon)$, task offloading policy $\mathbf{I}^{\mathbf{x}}(\varepsilon)$, mining node policy $\mathbf{I}^{\mathbf{b}}(\varepsilon)$, and bandwidth allocation $\mathcal{A}^{Bw}(\varepsilon)$. Concretely, the first and last components imply $\mathcal{A}^q(\varepsilon) = \{\eta_i(\varepsilon), i = 1,2,\ldots,Q\}$ and $\mathcal{A}^{Bw}(\varepsilon) = \{B_{w,i}(\varepsilon), i = 1,2,\ldots,W, i \neq w\}$, respectively. Consequently, the agent action $\mathfrak{a}(\varepsilon)$ is defined as

$$\mathfrak{a}(\varepsilon) \triangleq \{\mathcal{A}^q(\varepsilon), \mathbf{I}^{\mathbf{x}}(\varepsilon), \mathbf{I}^{\mathbf{b}}(\varepsilon), \mathcal{A}^{Bw}(\varepsilon)\} \tag{25}$$

3) Reward Function: In accordance with DRL, an agent is rewarded or punished for taking an action in a state, which guides the decision-making process towards optimal outcomes. When there is a current state $\mathfrak{s}_\varepsilon$, the agent samples an action $\mathfrak{a}_\varepsilon$ based on the policy function $\pi(\mathbb{A}|\mathbb{S}) = \mathcal{P}^\pi(\mathfrak{a}_\varepsilon|\mathfrak{s}_\varepsilon)$ and gets a reward $\mathfrak{r}_\varepsilon$. Here, we use the previously mentioned objective as the reward function to control the actions of the agent, which includes KGR, offloading latency and blockchain latency for maximum quantum internet security and user experience. Furthermore, the goal of the algorithm incorporates the consideration of rewards for the future, which implies that current rewards have the greatest impact on the current moment and decreasing impact on future moments. Therefore, denote $\mathfrak{G}_\varepsilon$ as the accumulated discount reward at the time slot ε, which is expressed as

$$\mathfrak{G}_\varepsilon = \mathfrak{r}_\varepsilon + \phi\mathfrak{r}_{\varepsilon+1} + \phi^2\mathfrak{r}_{\varepsilon+2} + \cdots = \sum_{k=0}^{\infty} \phi^k \mathfrak{r}_{\varepsilon+k} \tag{26}$$

where $\phi \in (0,1)$ means the discount factor and indicates that the current reward is decreasingly beneficial in the future. Based on the [47], [48], the state transition probability function provides a description of the dynamics of the environment, which represents how the environment responds to the actions of the agents and evolves to a new state. In this paper, this function can be defined as

$$\mathfrak{s}_{\varepsilon+1} = \hat{\mathcal{P}}_{sys}(\mathfrak{s}_{\varepsilon+1}|\mathfrak{s}_\varepsilon, \mathfrak{a}_\varepsilon) \tag{27}$$

Subsequently, the value function can be evaluated by the current state-action pair $(\mathfrak{s}, \mathfrak{a})$ and policy π according to the following equation

$$\mathbf{Q}^{\pi}(\mathfrak{s}, \mathfrak{a}) = \mathfrak{r}(\mathfrak{s}, \mathfrak{a}) + \phi \sum_{s^{\circ} \in \mathbb{S}} \mathcal{P}(\mathfrak{s}^{\circ}, \mathfrak{a}) \sum_{\mathfrak{a}^{\circ} \in \mathbb{A}} \pi(\mathfrak{s}^{\circ}, \mathfrak{a}^{\circ}) \mathbf{Q}^{\pi}(\mathfrak{s}^{\circ}, \mathfrak{a}^{\circ}) \tag{28}$$

In the proposed algorithm, the optimized policy can be formulated as

$$\pi^{*} = \underset{\pi}{\operatorname{argmax}} \mathbf{Q}^{\pi}(\mathfrak{s}, \mathfrak{a}), \forall \mathfrak{s} \in \mathbb{S} \tag{29}$$

Accordingly, the optimized action value function can be derived by

$$\mathbf{Q}^{*}(\mathfrak{s}, \mathfrak{a}) \triangleq \mathfrak{r}(\mathfrak{s}, \mathfrak{a}) + \mathfrak{z} \sum_{\mathfrak{s}^{\circ} \in \mathbb{S}} \mathcal{P}(\mathfrak{s}^{\circ}, \mathfrak{a}) \max_{\mathfrak{a}^{\circ}} \mathbf{Q}^{\pi}(\mathfrak{s}^{\circ}, \mathfrak{a}^{\circ}) \tag{30}$$

where $\mathfrak{z}$ is a hyperparameter that represents the learning rate during the parameter optimization of the system model. In the real VEC scenario, we utilize a one-step updating policy to rapidly approximate the optimal policy to quickly accommodate dynamically changing road conditions and instantaneous data. The equation (30) can be derived as

$$\mathbf{Q}(\mathfrak{s}_{\varepsilon}, \mathfrak{a}_{\varepsilon}) \leftarrow \mathfrak{z} \left(\phi \max_{\mathfrak{a}^{\circ}} \mathbf{Q}^{\pi}(\mathfrak{s}_{\varepsilon+1}, \mathfrak{a}^{\circ}) + \mathfrak{r}_{\varepsilon+1}\right) + (1 - \mathfrak{z}) \mathbf{Q}(\mathfrak{s}_{\varepsilon}, \mathfrak{a}_{\varepsilon}) \tag{31}$$

Let $\hat{y}_{\varepsilon}$ express the target Q-values, and then the formula can be written as

$$y_{\varepsilon} = \mathfrak{r}_{\varepsilon} + \phi(\mathbf{V}^{\pi}(\mathfrak{s}_{\varepsilon}^{\circ}, \mathbf{V}^{\pi}(\mathfrak{s}_{\varepsilon}^{\circ}; \varpi^{\circ *}(\varepsilon - 1)); \varpi^{\mathbf{Q}^{\circ}}(\varepsilon - 1))) \tag{32}$$

where $\mathbf{V}$ is the expected return under all possible state distributions, ϖ means the parameters of the target network and $\mathfrak{s}_{\varepsilon}^{\circ}$ represents the next state. Then, the loss Γ at layer i in the proposed TN-QDRL algorithm can be computed as

$$\Gamma^{(i)}(\varpi^{\mathbf{Q}}) = \frac{1}{\aleph} \sum_{\varepsilon}^{\aleph - 1} (\mathbf{V}^{\pi}(\mathfrak{s}_{\varepsilon}, \mathfrak{a}_{\varepsilon}; \varpi^{\mathbf{Q}}(\varepsilon - 1)) - \hat{y}_{\varepsilon})^{2} \tag{33}$$

where $\aleph$ denotes the total number of rounds of iteration.

B. Quantum States Representation

Based on the physics of quantum systems, states and actions in DRLs can be represented by quantum superposition states. The quantum superposition state is a special kind of quantum state that can contain multiple possible states and actions at the same time. This representation greatly extends the capabilities of traditional DRL, allowing it to handle complex learning and decision-making problems more efficiently. To better understand this representation, we can utilize the Dirac's representation to define the quantum superposition of a system, which adopts the symbol $|\psi\rangle$ to denote a quantum state. For a n-state system, its quantum superposition can be expressed as

$$|\psi\rangle = \sum_{i=0}^{n-1} \ell_i |i\rangle \tag{34}$$

where $\ell_0, \ell_1, \ldots \ell_{n-1}$ represent complex coefficients with normalization condition of $\sum_{i=0}^{n-1} |\ell_i|^2 = 1$. Particularly, for qubits of the quantum computer, take N equal to 2.

In real-time dynamic VEC environments, traditional Q-table face bottlenecks in storage and computational resources due to large state spaces, leading to increased latency and higher blocking probability. Although quantum computing can accomplish algorithms that cannot be realized by conventional computers in an acceptable time, QDRL fails to fully exploit the potential capabilities of quantum due to the limitation of the number of available quantum bits in the NISQ era. Therefore, we employ a tensor preprocessing-based quantum algorithm that utilizes a TN to process dimensional inputs which exceed the number of quantum bits. This algorithm enables the full capability of QDRL to be exploited, reduces the exploration time, significantly improves the performance and efficiency of the system, and optimizes the task processing.

1) Tensor Network: The domain of quantum many-body physics pioneered the invention of the TN technique [49], [50]. Therein, matrix product state (MPS) stands for a special kind of one-dimensional TN, which is realized by resolving a large tensor into a sequence of matrices. In general, the quantum state of M qubits can be expressed as

$$|\Psi\rangle = \sum_{l_1} \sum_{l_2} \cdots \sum_{l_j} A_{l_1, l_2 \cdots l_j} |l_1\rangle \otimes |l_2\rangle \otimes \cdots \otimes |l_j\rangle \tag{35}$$

where each basis state $|l_1\rangle \otimes |l_2\rangle \otimes \cdots \otimes |l_j\rangle$ has an amplitude $A_{l_1, l_2 \cdots l_j}$. As the quantity of qubits is added, the population of $A_{l_1, l_2 \cdots l_j}$ increases exponentially, which can make it extremely difficult to compute and memorize on classical computers. Fortunately, MPS can efficiently denote these amplitudes as matrix multiplications according to [51]:

$$A_{l_1, l_2 \cdots l_j} = \sum_{\zeta_1} \sum_{\zeta_2} \cdots \sum_{\zeta_j} M^{1}_{l_1 \zeta_1} M^{2}_{\zeta_1 l_2 \zeta_2} M^{3}_{\zeta_2 l_3 \zeta_3} \cdots M^{\jmath}_{\zeta_{\jmath - 1} l_j} \tag{36}$$

where there are $\jmath$ matrices and ζ_i means the virtual indices. The adjustable hyperparameter of the bond dimension for each matrix is χ.

2) MPS-based Operation: In the proposed TN-QDRL, we employed the MPS-based feature extractor to reduce the dimensionality of the given vector $\overrightarrow{\mathfrak{v}}$ with the feature map $\left|\Upsilon(\overrightarrow{\mathfrak{v}})\right\rangle$, which is denoted as

$$\overrightarrow{\mathfrak{v}} \longrightarrow \left|\Upsilon(\overrightarrow{\mathfrak{v}})\right\rangle = o(\mathfrak{v}_1) \otimes o(\mathfrak{v}_2) \otimes \cdots \otimes o(\mathfrak{v}_j) \tag{37}$$

where each $\mathfrak{v}_j$ meshed into a $\mho$-dimensional vector with the function $o(\cdot)$, which is called the local dimension. $\mho = 2$ is utilized to ensure the convergence speed and stability of the proposed algorithm based on [52], and can be written as

$$o(\mathfrak{v}_j) = \begin{bmatrix} 1 - \mathfrak{v}_j \\ \mathfrak{v}_j \end{bmatrix} \tag{38}$$

Here, the input vector $\mathfrak{v}_j$ represents the state of the system observed by agents in QIB-VEC. Then, the output entered into the VQC can be inferred from equation (37):

$$\overrightarrow{\mathfrak{v}} \longrightarrow \left|\Upsilon(\overrightarrow{\mathfrak{v}})\right\rangle = \begin{bmatrix} 1 - \mathfrak{v}_1 \\ \mathfrak{v}_1 \end{bmatrix} \otimes \begin{bmatrix} 1 - \mathfrak{v}_2 \\ \mathfrak{v}_2 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 - \mathfrak{v}_j \\ \mathfrak{v}_j \end{bmatrix} \tag{39}$$

The dimensionality reduction of the original vector with a trainable MPS can be denoted as

$$\Theta(\overrightarrow{\mathfrak{v}}) = \sum_{l_1}\sum_{l_2}\cdots\sum_{l_j} A_{l_1,l_2\cdots l_j} o(\mathfrak{v}_1)_{l_1} \otimes o(\mathfrak{v}_2)_{l_2} \otimes \cdots \otimes o(\mathfrak{v}_j)_{l_j} \tag{40}$$

where $\mathfrak{v}_0, \mathfrak{v}_1, \cdots, \mathfrak{v}_j \in \{0, 1\}$, and the compressed dimension can be specified based on $A_{l_1,l_2\cdots l_j}$ with the equation (36). As shown in the Fig. 3, the blue nodes denote the encoded feature-mapped input and the green nodes represent the trainable MPS.

3) Quantum Reinforcement Learning: By virtue of the superposition principle in quantum information theory, the state of the system observed and the action taken by the agent can be presented as

$$|\mathfrak{S}_\varepsilon\rangle = \sum_i \eth_i^s |\mathfrak{s}_\varepsilon\rangle \tag{41}$$

$$|\mathfrak{A}_\varepsilon\rangle = \sum_j \eth_j^a |\mathfrak{a}_\varepsilon\rangle \tag{42}$$

where $\eth_i^s$ and $\eth_j^a$ are amplitudes, which fulfill the normalization condition $\sum_i |\eth_i^s|^2 = 1$ and $\sum_j |\eth_j^a|^2 = 1$, respectively. Assume that the number of components of the state space and action space are D_a and D_b, where $D_a = 2^\alpha$, $D_b = 2^\beta$, and α and β express the number of qubits. Considering the case of division, the conditions $D_a \le 2^\alpha \le 2D_a$ and $D_b \le 2^\beta \le 2D_b$ need to be fulfilled. Then, equations of the superposition in state space and action space can be written as

$$\left|\mathfrak{S}_\varepsilon^{(D_a)}\right\rangle \longrightarrow \left|\mathfrak{S}_\varepsilon^{(\alpha)}\right\rangle = \sum_{\mathcal{S}=\overbrace{00\cdots0}^{\alpha}}^{\overbrace{11\cdots1}^{\alpha}} \eth_{\mathfrak{s}} |\mathcal{S}\rangle, \sum_{\mathcal{S}=00\cdots0}^{11\cdots1} |\eth_{\mathfrak{s}}|^2 = 1 \tag{43}$$

$$\left|\mathfrak{A}_\varepsilon^{(D_b)}\right\rangle \longrightarrow \left|\mathfrak{A}_\varepsilon^{(\beta)}\right\rangle = \sum_{\mathcal{A}=\overbrace{00\cdots0}^{\beta}}^{\overbrace{11\cdots1}^{\beta}} \eth_{\mathfrak{a}} |\mathcal{A}\rangle, \sum_{\mathcal{A}=00\cdots0}^{11\cdots1} |\eth_{\mathfrak{a}}|^2 = 1 \tag{44}$$

where $\eth_{\mathfrak{s}}$ and $\eth_{\mathfrak{a}}$ indicate the probability amplitudes.

C. Grover's Search Algorithm

Grover's search algorithm is a significant algorithm in the field of quantum computing. It is particularly effective at solving the problem of searching an unsorted database. While a classical computer requires $O(N)$ time complexity to find a specific element in an unsorted database, this algorithm reduces the time complexity to $O(\sqrt{N})$. When an agent makes a decision, it chooses an action to perform that is relevant to the current state. Specifically, after a measurement, the state collapses to $|\mathcal{A}\rangle$ with probability $|\eth_{\mathfrak{a}}|^2$. The main idea of the Grover algorithm is to make the amplitude of the action with a high reward larger through iterative optimization, thus enhancing the probability of taking this action. In this paper, the policy can be characterized as $\pi : S \times \cup_{x\in S} A_{(x)}$, and the corresponding action in state S is denoted as $\Lambda(\mathfrak{A}_S) = \left|\mathfrak{A}_\epsilon^{(D_b)}\right\rangle = \left|\mathfrak{A}_\epsilon^{(\beta)}\right\rangle$. The following main phases describe the quantum circuits of the Grover operator in the proposed algorithm.

1) Initialization: For the superposition, the 2^β possible eigenspaces together form $\left|\mathfrak{A}_\epsilon^{(\beta)}\right\rangle$ based on the Hadamard transform below.

$$\begin{aligned}\left|\mathfrak{A}_\varepsilon^{(\beta)}\right\rangle &= H^{\otimes\beta}\left|\overbrace{00\cdots0}^{\beta}\right\rangle \\ &= \frac{1}{\sqrt{2^\beta}}|00\cdots0\rangle + \cdots + |11\cdots1\rangle \\ &= \frac{1}{\sqrt{2^\beta}}\sum_{i=0}^{2^\beta-1}|\mathcal{A}\rangle\end{aligned} \tag{45}$$

where $1/\sqrt{2^\beta}$ is the initial value of the amplitude. $H^{\otimes\beta}$ means the Hadamard transform, which is expressed as

$$H^{\otimes\beta} = \frac{1}{\sqrt{2^\beta}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}^{\otimes\beta} \tag{46}$$

Subsequently, the initial state can be derived as

$$\begin{aligned}\Lambda(\mathfrak{A}_S) &= \left|\mathfrak{A}_{\epsilon,\epsilon=0}^{(\beta)}\right\rangle \\ &= \frac{1}{\sqrt{2^\beta}}\sum_{i=0}^{2^\beta-1}|\mathcal{A}\rangle \\ &= \frac{1}{\sqrt{2^\beta}}(|\mathcal{A}\rangle + \sum_{i\neq\mathcal{A}}|i\rangle) \\ &= \frac{1}{\sqrt{2^\beta}}|\mathcal{A}\rangle + \frac{\sqrt{2^\beta-1}}{\sqrt{2^\beta}}\cdot\frac{\sum_{i\neq\mathcal{A}}|i\rangle}{\sqrt{2^\beta-1}} \\ &= \frac{1}{\sqrt{2^\beta}}|\mathcal{A}\rangle + \sqrt{\frac{2^\beta-1}{2^\beta}}|\mathcal{A}^\perp\rangle\end{aligned} \tag{47}$$

where $\left|\mathcal{A}^\perp\right\rangle = (|0\rangle - |1\rangle)/\sqrt{2^\beta}$. Assume that $\left\langle\mathcal{A}|\mathcal{A}_0^\beta\right\rangle = 1/\sqrt{2^\beta} \equiv \sin\partial$, and then the equation (47) can be simplified and expressed as

$$\Lambda(\mathfrak{A}_\varepsilon) = \left|\mathfrak{A}_0^{(\beta)}\right\rangle = \sin\partial|\mathcal{A}\rangle + \cos\partial\left|\mathcal{A}^\perp\right\rangle \tag{48}$$

2) Oracle Application: Quantum oracles are black-box functions, capable of performing a particular operation, which can be represented as a unitary operator. In TN-QDRL, the oracle denoted by $\mathcal{O}^q$ can be written as

$$|\psi_1\rangle : \left|\mathfrak{A}_\epsilon^{(\beta)}\right\rangle \overset{\mathcal{O}^q}{\rightarrow} \frac{1}{\sqrt{2^\beta}}\sum_{i=0}^{2^\beta-1}(-1)^{\Lambda(\mathcal{A})}|\mathcal{A}\rangle\left|\mathcal{A}^\perp\right\rangle \tag{49}$$

3) Diffusion Operator Application: For the Grover's operation, we define the operator as $\mathcal{G}_q = \mathcal{O}^q(2\left|\mathfrak{A}_\varepsilon^{(\beta)}\right\rangle\left\langle\mathfrak{A}_\varepsilon^{(\beta)}\right| - I)$, where I means the unit matrix. Then, Grover iteration process can be expressed by adopting the unitary transformation $\mathcal{G}_q$:

$$|\psi_2\rangle : |\psi_1\rangle \overset{\mathcal{G}_q}{\rightarrow} [(2\left|\mathfrak{A}_\varepsilon^{(\beta)}\right\rangle\left\langle\mathfrak{A}_\varepsilon^{(\beta)}\right| - I)|\psi_1\rangle]^\tau \approx \mathfrak{A}_S\frac{|0\rangle - |1\rangle}{\sqrt{2^\beta}} \tag{50}$$

where τ represents the Grover iteration time throughout the entire process. By utilizing quantum superposition and quantum interference principles, $\left|\mathfrak{A}_\epsilon^{(\beta)}\right\rangle$ is updated iteratively. For a complete practical application and theoretical underpinnings, the detailed breakdown is presented in Algorithm 2.

Algorithm 2: Proposed TN-QDRL Algorithm for QIB-VEC

Input: TN Θ, classic system state $\mathfrak{s}_\varepsilon$, available action $\mathfrak{a}_\varepsilon$ and Q-value function $\mathbf{Q}$

Output: Optimal resource allocation action $\mathfrak{A}_\varepsilon$

Use feature map $|\Upsilon(\cdot)\rangle$ to decompose the system state $\mathfrak{s}_\varepsilon$ into dimensionality reduced vectors.

Obtain the state of the system after down scaling $\mathfrak{s}_\varepsilon \longleftarrow \Theta(\mathfrak{s}_\varepsilon)$.

Map the reduced system state to a quantum state $\left|\mathfrak{S}_\varepsilon^{(D_a)}\right\rangle \longrightarrow \sum_{\mathcal{S}=00\cdots0}^{\overbrace{11\cdots1}^{\alpha}} \eth_{\mathfrak{s}} |\mathcal{S}\rangle$.

Map the action to a quantum state $\left|\mathfrak{A}_\varepsilon^{(D_b)}\right\rangle \longrightarrow \sum_{\mathcal{A}=00\cdots0}^{\overbrace{11\cdots1}^{\beta}} \eth_{\mathfrak{a}} |\mathcal{A}\rangle$.

Initialize $T = 0$.

for $T \leq T_{max}$ **do**

  Sample an action with amplitude as probability $\eth_{\mathfrak{a}}$.

  if *the action* $|\mathcal{A}\rangle$ *is obtained* **then**

    Attain the instant reward $\mathfrak{r}_\varepsilon$.

    Observe the next state $\mathfrak{s}_{\varepsilon+1}$ of the system after taking $|\mathcal{A}\rangle$.

    Calculate the reduced state $\mathfrak{s}_{\varepsilon+1} \longleftarrow \Theta(\mathfrak{s}_{\varepsilon+1})$.

    Get the quantum state $\left|\mathfrak{S}_{\varepsilon+1}^{(D_a)}\right\rangle$.

    Update the system value iteratively according to (31).

    Repeat the Grover iteration process τ times based on the operator $\mathcal{G}_q$.

  end

end

V. Experimental Analysis and Discussions

In this section, we first express the simulation setup, detailing the parameters and configurations used to create the simulation environment. Then, we introduce the four evaluation metrics for experiments. Following this, we discuss the simulation results, which provides a comprehensive analysis of the data obtained.

A. Simulation Setup

In this paper, we utilize simulation results to comprehensively analyze the performance of joint RA schemes integrating QKD and computational processes. The entire simulation process is conducted on a Windows 11 Enterprise operating system, running on a machine equipped with a 12th Gen Intel (R) Core (TM) i7-12700KF 3.60 GHz CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 4060 Ti GPU, ensuring high computational efficiency. For quantum computing simulations, we leverage the IBM Quantum Experience platform alongside Qiskit (v0.43), a Python-based quantum computing framework, for modeling quantum operations under realistic NISQ device constraints. The simulations are further supported by Python (v3.10), with key libraries including NumPy (v1.24.2) for numerical operations, TensorFlow (v2.13) for implementing deep reinforcement learning models, and Matplotlib (v3.7) for visualizing the results. Vehicular traffic and mobility patterns are simulated using SUMO (v1.16.0), which provides dynamic topologies and realistic urban traffic scenarios. These configurations, combined with diverse task profiles and constrained quantum resources, ensure the robustness and relevance of the simulation results, validating the efficiency and scalability of the proposed TN-QDRL algorithm in real-world VEC applications.

The parameters that we set in Table II encompass various critical aspects of the simulation environment according to [7] and [47], including initial conditions, boundary settings, and key variable thresholds, all of which are essential for replicating real-world scenarios with high fidelity. $\kappa_{i,j}^{qbe}$, $\kappa_{i,j}^{ent}$, ϖ_I, and η_i are derived from the current performance ranges of quantum communication networks. c_x^v and c_z^e reflect the computational performance of in-vehicle devices and edge servers. Task-related parameters including the required CPU cycles and data size are designed to mimic typical workloads in vehicular networks. Wi-Fi and cellular bandwidths, σ_v and $\gamma_x^{v,m}$ are set based on standard vehicular communication protocols and typical noise environments. ϕ, $\ni$, ξ_1, and ξ_2 are fine-tuned according to standard practices in RL to ensure convergence and stability. Blockchain parameters, including s^b and t_b^g, are chosen to reflect typical blockchain processing capabilities. The maximum tolerable delay was set to meet the responsiveness requirements of vehicular network applications.

To compare the experimental results more obviously, we establish five different variant schemes of the proposed TN-QDRL:

1) **RALC** (*RA scheme with only local computing*) [53]: In this scheme, all computational tasks are executed locally on vehicles without any task offloading to edge servers. RALC is limited by the computational capabilities of individual vehicles by eliminating task offloading, which can lead to suboptimal performance under heavy task loads or stringent latency requirements.
2) **RARS** (*RA scheme with random selection*) [54]: This scheme allocates resources randomly without employing any optimization or intelligent decision-making. RA decisions are made without considering task priorities, network conditions, or resource availability. As a result, this configuration helps to demonstrate the importance of using intelligent optimization algorithms for achieving efficient and balanced resource utilization.
3) **RARL** (*RA scheme with classic RL algorithm*) [55]: This scheme employs a Q-learning algorithm to manage RA. In contrast to TN-QDRL, which integrates QRL with tensor network preprocessing to address high-dimensional state spaces, RARL relies solely on a classic RL approach. This scheme evaluates the performance differences between traditional and quantum-enhanced techniques, particularly in terms of convergence speed, resource efficiency, and adaptability in dynamic VEC environments.
4) **RANQKD** (*RA scheme without QKD*) [56]: This scheme removes QKD from the RA process, resulting in a

system without secure quantum communication. By disregarding quantum key resources for encryption, the scheme assesses the importance of QKD in ensuring secure and reliable communication, as well as its impact on system performance and RA efficiency.

5) **RAFBZ** (*RA scheme with the fixed block size*) [57]: In this scheme, resources are allocated in predetermined, fixed-sized blocks regardless of the specific needs of individual tasks. This static allocation strategy ignores the dynamic characteristics of vehicular applications, such as varying task sizes and deadlines. Through comparing RAFBZ to TN-QDRL with a dynamic adaptive allocation strategy, we demonstrate the advantages of tailoring resource allocation to specific task requirements.

Each of these schemes is specifically designed to isolate and analyze the effects of key features in TN-QDRL, such as the use of edge computing, optimization algorithms, quantum reinforcement learning, QKD, and dynamic resource allocation. By comparing the proposed method with these variants, we aim to demonstrate the advantages and robustness of TN-QDRL in a comprehensive and systematic manner.

B. Evaluation Metrics

To comprehensively evaluate the performance of the proposed approach, we utilize four key metrics, each addressing a specific aspect of the functionality of the proposed system:

1) System Returns: This metric captures the overall performance of the reinforcement learning-based resource allocation strategy. By analyzing the cumulative rewards obtained, it reflects the ability of the system to optimize task scheduling and improve the QoE for users.

2) Average Latency: This metric measures the responsiveness of the system, specifically focusing on the time delay involved in task offloading and processing. It is critical for latency-sensitive applications in VEC.

3) KGR: As a measure of security performance, KGR evaluates the speed and efficiency of QKD in generating secure keys for communication. It reflects the ability of the system to maintain secure operations under dynamic vehicular conditions.

4) Average Resource Utilization: This metric assesses how effectively the system utilizes available computational, storage, and quantum resources. It highlights the balance between maximizing resource usage and minimizing inefficiencies, ensuring sustainable operation in resource-constrained environments. At the same time, this indicator also reflects the cost-effectiveness of the resources.

Based on these four indicators, we have made the following overall assessment of the QIB-VEC.

C. Simulation Results

This Fig. 4 illustrates the convergence results of the TN-QDRL algorithm under different learning rates on the QIB-VEC environment. Four learning rates are compared: 0.1, 0.01, 0.001, and 0.0001. The red curve (LR=0.1) shows rapid convergence and stabilizes around 200 returns after approximately 100 episodes, indicating fast adaptation. The green curve (LR=0.01) stabilizes at around 175 returns after about 150 episodes, suggesting a slower rate. The blue curve (LR=0.001) converges more slowly, achieving moderate returns after around 400 episodes. The orange curve (LR=0.0001) displays very slow convergence with low returns, indicating insufficient learning. Overall, LR=0.1 yields better performance in terms of returns and convergence speed, while lower learning rates show slower and less effective learning.

Fig. 5 shows the system returns of different schemes on QIB-VEC over 800 episodes. TN-QDRL converges quickly with its complete algorithmic setup and remains remarkably steady at high returns. RARL also performs well and plateaus around 190 returns, which benefits from traditional Q-learning. RARS and RAFBZ are limited by the random computation selection and fixed block size, respectively, and have slow convergence and medium returns. RALC has lower returns

TABLE II
KEY PARAMETERS IN THE SIMULATION

Parameter	Value
QKD error rate $\kappa_{i,j}^{qbe}$	$[0.01, 0.05, 0.1]$ bps
Quantum entanglement rate $\kappa_{i,j}^{ent}$	$[10, 20, 50]$ EPRpairs/s
Interference visibility ϖ_I	$[0.8, 0.85, 0.9, 0.95, 1.0]$
Detector Efficiency η_i	$[0.8, 0.9, 1.0]$
Wi-Fi Bandwidth from vehicles to RSUs	35 MHz
Bandwidth for cellular	35 MHz
Required CPU cycles of computing tasks	$[0.5, 1.8, 3.0]$ Gcycles/s
AWGN powewr σ_v	-174 dBm
SINR threshold $\gamma_x^{v,m}$	18 dB
Processing density for blocks ρ^b	737.5 cycle/bit
Tolerable maximum delay	$[0.2, 1]$ seconds
Transmission power $p_x^{v,m}$	$[1, 5, 10, 20]$ W
Local computing capability c_x^v	0.5 GHz
Edge computing capability c_z^e	2.5 GHz
Each data size of vehicles $s_{x,z}^v$	$0 \sim 60$ MB
Transmission rate of data $\nu_x^{v,m}$	15×10^6 bit/s
Weighting factor and meshing factor ξ_1, ξ_2	$[0.4, 0.5]$
Discount factor ϕ	[0.9, 0.95, 0.99]
Learning rate ϑ	[0.001, 0.01, 0.05]
Block size s^b	2 KB
Block generation time t_b^g	$[10, 20, 30]$ seconds

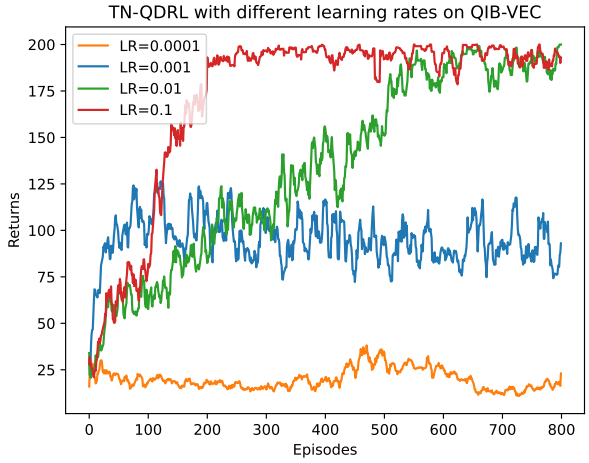

Fig. 4. Convergence results under different learning rates: 1) returns with learning rate = 0.1; 2) returns with learning rate = 0.01; 3) returns with learning rate = 0.001; 4) returns with learning rate = 0.0001.

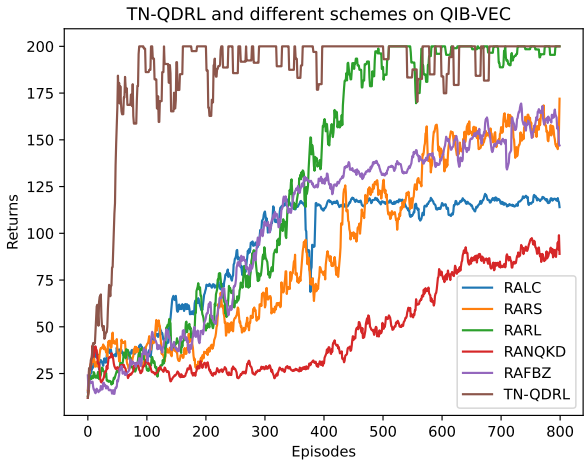

Fig. 5. System returns of different schemes.

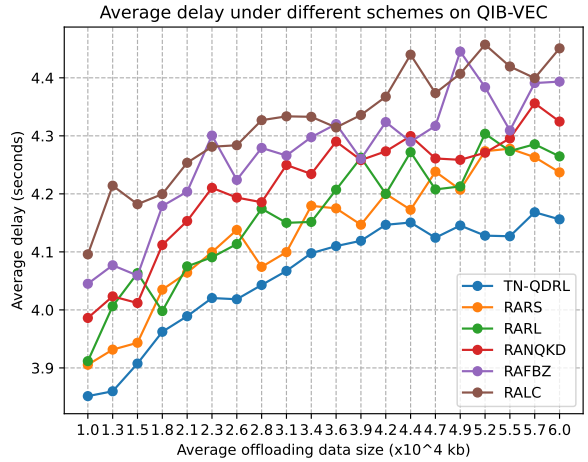

Fig. 6. Average delays of different schemes with various offloading data sizes.

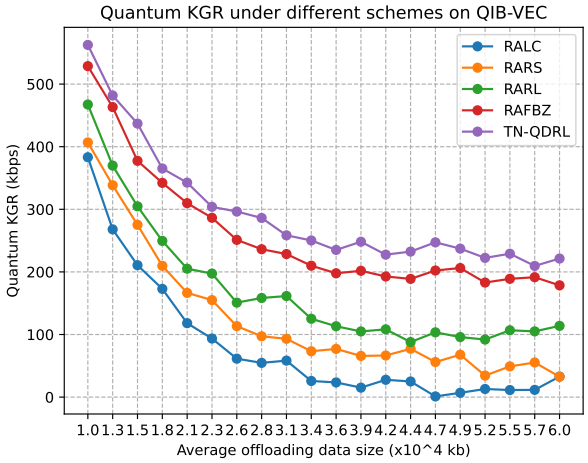

Fig. 7. Average KGR of different schemes with various offloading data sizes.

due to the localized computation, but the algorithm still works and can converge. RANQKD has the poorest performance and stabilizes below 100, since the QKD module is not considered in the system rewards.

Fig. 6 demonstrates the average delay under different schemes on QIB-VEC with various offloading data sizes. Our proposed algorithm consistently achieves the lowest average latency, since it optimizes the Grover's iterations to manage resources efficiently. RARL and RARS have and more variable delays due to random resource allocation and the slower conventional algorithm. RAFBZ and RANQKD experience higher latency caused by fixed block size allocation and lack of quantum key resources, respectively. RALC can only utilize limited local resources and spends most of its time on computation, which results in higher latency.

Fig. 7 shows that our proposed algorithm consistently outperforms the other schemes, starting high and decreasing as the offloading data size increases, but maintaining the highest KGR overall. RALC has the worst performance, while RARS and RARL behave similarly, with RARL having a slight edge. RAFBZ shows a more stable KGR across different data sizes but falls behind TN-QDRL. These trends suggest that TN-QDRL efficiently manages quantum key resources and prioritizes offloading, whereas the local-only scheme struggle with scalability and optimization of KGR due to low transmission volumes.

Fig. 8 demonstrates the TN-QDRL has the lowest average delay across all vehicle numbers, indicating its efficiency in handling increased traffic. RARS, RARL, and RANQKD exhibit similar delay patterns, with a gradual increase in delay as the number of vehicles grows. RAFBZ and RALC consistently show the highest delays, particularly as the number of vehicles increases, since fixed block allocation and local-only computing less effective in managing increased load.

Fig. 9 shows that the proposed TN-QDRL algorithm consistently maintains a high KGR, even as the number of vehicles increases, demonstrating its robustness in handling dynamic and resource-intensive scenarios. In contrast, RARL and RAFBZ achieve moderate KGR levels but are limited by slower adaptability and lack of flexibility in resource allocation. RARS and RALC exhibit the lowest KGR as the vehicular density increases, primarily due to the absence of optimization strategies and reliance on local computing resources, respectively. These results highlight the effectiveness of TN-QDRL in optimizing resource allocation while ensuring secure communication in high-demand vehicular networks.

Fig. 10 illustrates the quantum resource cost across multiple episodes for three algorithms: TN-QDRL (ours), Anserre et al. [47], and Wang et al. [7]. The algorithms are selected for comparison due to their relevance in addressing similar challenges in quantum resource management and their demonstrated effectiveness in previous studies. This allows for a comprehensive evaluation of the performance of TN-QDRL against established approaches. The results highlight notable performance differences in terms of resource efficiency and stability. The proposed TN-QDRL algorithm consistently exhibits the lowest quantum resource cost throughout the episodes. This advantage is largely attributed to the integration of TN preprocessing, which effectively reduces the dimension-

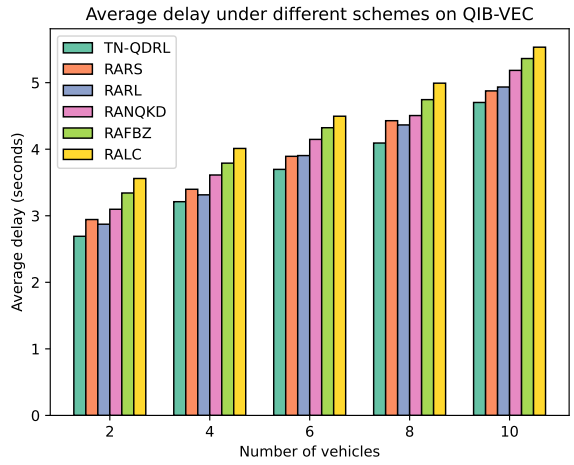

Fig. 8. Average delays of different schemes with various numbers of vehicles.

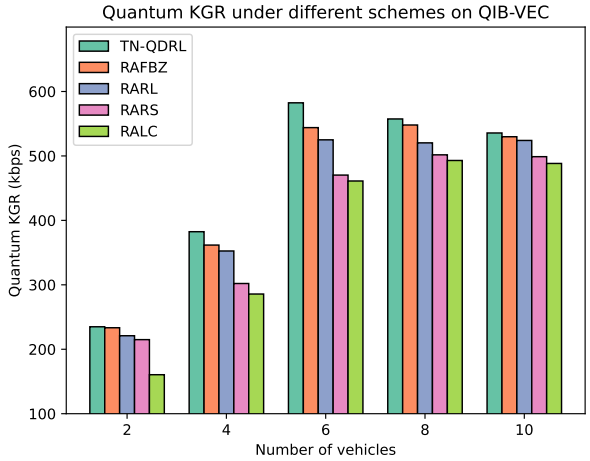

Fig. 9. Average KGR of different schemes with various numbers of vehicles.

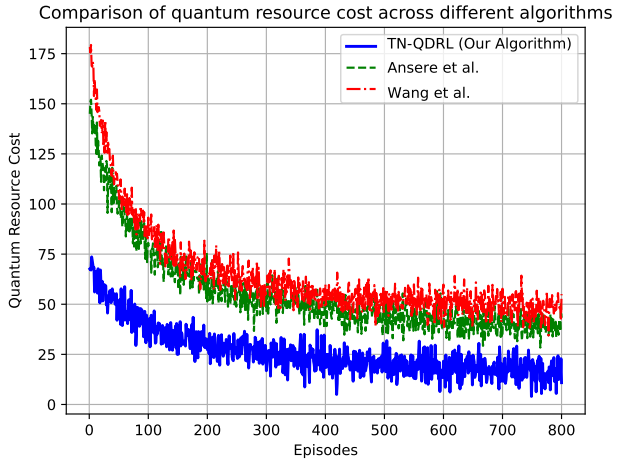

Fig. 10. Average resource utilization of different quantum algorithms.

ality of the state space while preserving essential information. This preprocessing step not only minimizes the number of qubits required, but also optimizes the quantum operations, leading to more efficient utilization of limited resources on NISQ devices. Furthermore, TN-QDRL maintains stability in resource consumption across episodes, indicating its robustness in adapting to dynamic vehicular edge computing (VEC) scenarios. In contrast, the algorithms by Anserre et al. [47] and Wang et al. [7] demonstrate significantly higher and more variable resource costs, especially in the initial episodes. This variability can be attributed to their reliance on less optimized quantum encoding and processing techniques, which may not fully leverage the advantages of tensor networks or other resource-efficient methods. In the case of Anserre et al., the fluctuations suggest a learning process that struggles to converge quickly, resulting in inconsistent resource usage. Similarly, the algorithm by Wang et al. exhibits initial inefficiencies, likely due to suboptimal adaptation mechanisms when faced with changing environmental conditions in VEC. The performance differences underline the superiority of TN-QDRL in resource efficiency and adaptability. By leveraging the quantum properties of superposition and entanglement in combination with tensor network preprocessing, TN-QDRL effectively handles the complexities of high-dimensional state spaces, ensuring a balance between resource conservation and robust decision-making. This capability is particularly important for NISQ devices, where hardware limitations impose strict constraints on the number of available qubits and the fidelity of quantum operations. The results demonstrate that TN-QDRL not only minimizes resource cost but also achieves steady and reliable performance, making it a highly practical choice for real-world applications.

VI. Conclusion

A. Limitations and Assumptions

The TN-QDRL algorithm has minor limitations that highlight areas for improvement. Current quantum hardware in the NISQ era imposes constraints such as limited qubit fidelity, coherence time, and high error rates, which can hinder the practical implementation of the algorithm. Additionally, the TN-based dimensionality reduction may lead to the loss of critical features in extremely complex environments, potentially affecting decision quality.

There are several assumptions that shape its theoretical foundation and application scope. Firstly, it assumes the availability of near-term quantum devices with sufficient qubit quality, coherence, and quantity to perform operations such as superposition, Grover's search, and quantum state encoding efficiently. Then, the algorithm presumes that quantum cryptographic techniques will be leveraged in systems such as quantum blockchain to ensure secure communication and resource allocation. It is also assumed that blockchain systems will eventually adopt post-quantum cryptographic algorithms to remain secure against adversaries with advanced quantum computing capabilities.

B. Significance and Impact

The achieved results represent a significant step forward in integrating VEC with quantum internet technologies, ad-

dressing critical challenges in security, efficiency, and scalability. The proposed TN-QDRL algorithm demonstrates notable improvements in resource allocation efficiency and response time, enabling effective operation in complex and dynamic vehicular scenarios while optimizing the use of limited quantum resources. By incorporating QKD and blockchain, the system ensures secure and reliable communication, which is essential for privacy-sensitive and latency-critical applications such as autonomous driving and real-time navigation. This work not only highlights the practical potential of quantum-enhanced methods in VEC, but also establishes a strong foundation for expanding quantum technologies into broader IoT and edge computing applications.

Furthermore, the scalability of the proposed TN-QDRL algorithm is a critical aspect of its significance, as it demonstrates the ability to effectively manage complex resource allocation tasks in large-scale vehicular edge computing scenarios. Through utilization of TN-preprocessing, efficient amplitude encoding, and quantum probabilistic search strategies, the algorithm efficiently handles the increased dimensionality and computational demands associated with larger networks. Moreover, its stability in performance metrics, such as convergence speed and resource consumption, highlights its adaptability to real-world environments. This scalability ensures that TN-QDRL can meet the demands of future intelligent transportation systems, where the number of connected vehicles and tasks will continue to grow exponentially.

C. Summary and Future Work

In this study, we explored the integration of QKD and blockchain technologies to enhance the security and efficiency of VEC systems. To address the critical need for effective RA in latency-sensitive tasks, we introduced a tensor preprocessing-based quantum-inspired reinforcement learning algorithm. Leveraging amplitude encoding, quantum superposition, and entanglement, this algorithm manages complex Markov decision processes in multi-dimensional state spaces. Our TN-QDRL algorithm incorporates an optimized search strategy using probabilistic quantum state transformations and an improved Grover's algorithm. Simulation results demonstrate that TN-QDRL achieves faster convergence in high-dimensional VEC scenarios and significantly reduces quantum resource consumption compared to existing benchmarks. By bridging quantum computing and quantum communication, our study provides a novel direction for future research, highlighting the potential of quantum-enhanced approaches to overcome current limitations and improve the performance and security of VEC systems.

Future research could focus on addressing scalability and robustness issues in large-scale and dynamic vehicular networks, as well as integrating hybrid classical-quantum systems to improve resource efficiency. Additionally, adaptive algorithms that account for real-world challenges are essential, such as qubit noise and variable latency. Exploring broader applications in other domains and optimizing for next-generation quantum hardware can further extend the practical relevance of the proposed approach.

REFERENCES

[1] S. Raza, S. Wang, M. Ahmed, M. R. Anwar *et al.*, "A survey on vehicular edge computing: architecture, applications, technical issues, and future directions," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[2] R. Meneguette, R. De Grande, J. Ueyama, G. P. R. Filho, and E. Madeira, "Vehicular edge computing: Architecture, resource management, security, and challenges," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–46, 2021.

[3] O. N. Nezamuddin, C. L. Nicholas, and E. C. dos Santos, "The problem of electric vehicle charging: State-of-the-art and an innovative solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4663–4673, 2021.

[4] M. Li, M. Zhang, L. Zhu, Z. Zhang, M. Conti, and M. Alazab, "Decentralized and privacy-preserving smart parking with secure repetition and full verifiability," *IEEE Transactions on Mobile Computing*, 2024.

[5] T. Chen, X.-P. Zhang, J. Wang, J. Li, C. Wu, M. Hu, and H. Bian, "A review on electric vehicle charging infrastructure development in the uk," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 2, pp. 193–205, 2020.

[6] XuanWen and H. M. Sun, "Parking cooperation-based mobile edge computing using task offloading strategy," *Journal of Grid Computing*, vol. 22, no. 1, p. 8, 2024.

[7] D. Wang, B. Song, P. Lin, F. R. Yu, X. Du, and M. Guizani, "Resource management for edge intelligence (ei)-assisted iov using quantum-inspired reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12 588–12 600, 2021.

[8] S. Xia, Z. Yao, Y. Li, and S. Mao, "Online distributed offloading and computing resource management with energy harvesting for heterogeneous mec-enabled iot," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6743–6757, 2021.

[9] B. Liang, F. Wang, and B. Ran, "Optimizing roadside unit deployment in vanets: A study on consideration of failure," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[10] W. Fan, Y. Su, J. Liu, S. Li, W. Huang, F. Wu, and Y. Liu, "Joint task offloading and resource allocation for vehicular edge computing based on v2i and v2v modes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4277–4292, 2023.

[11] Y. Xia, H. Zhang, X. Zhou, and D. Yuan, "Location-aware and delay-minimizing task offloading in vehicular edge computing networks," *IEEE Transactions on Vehicular Technology*, 2023.

[12] Y. Qian, J. Wu, R. Wang, F. Zhu, and W. Zhang, "Survey on reinforcement learning applications in communication networks," *Journal of Communications and Information Networks*, vol. 4, no. 2, pp. 30–39, 2019.

[13] X. Chen and G. Liu, "Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 843–10 856, 2021.

[14] S. Vaudenay, *A classical introduction to cryptography: Applications for communications security*. Springer Science & Business Media, 2005.

[15] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[16] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of modern physics*, vol. 81, no. 3, p. 1301, 2009.

[17] H. Weyl, "Quantenmechanik und gruppentheorie," *Zeitschrift für Physik*, vol. 46, no. 1, pp. 1–46, 1927.

[18] V. Bužek and M. Hillery, "Quantum copying: Beyond the no-cloning theorem," *Physical Review A*, vol. 54, no. 3, p. 1844, 1996.

[19] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, p. eaam9288, 2018.

[20] M. Xu, D. Niyato, Z. Yang, Z. Xiong, J. Kang, D. I. Kim, and X. Shen, "Privacy-preserving intelligent resource allocation for federated edge learning in quantum internet," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 142–157, 2022.

[21] Y. Cao, Y. Zhao, J. Li, R. Lin, J. Zhang, and J. Chen, "Multi-tenant provisioning for quantum key distribution networks with heuristics and reinforcement learning: A comparative study," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 946–957, 2020.

[22] Y. Liu, W.-J. Zhang, C. Jiang, J.-P. Chen, C. Zhang, W.-X. Pan, D. Ma, H. Dong, J.-M. Xiong, C.-J. Zhang *et al.*, "Experimental twin-field quantum key distribution over 1000 km fiber distance," *Physical Review Letters*, vol. 130, no. 21, p. 210801, 2023.

[23] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press Cambridge, 2001, vol. 2.

[24] M. Kumar, U. Dohare, S. Kumar, and N. Kumar, "Blockchain based optimized energy trading for e-mobility using quantum reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 5167–5180, 2023.

[25] G. S. Kim, J. Chung, and S. Park, "Realizing stabilized landing for computation-limited reusable rockets: A quantum reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, 2024.

[26] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, vol. 14, no. 4, pp. 352–375, 2018.

[27] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang, "Resource allocation and consensus of blockchains in pervasive edge computing environments," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3298–3311, 2021.

[28] S. Yang, J. Tan, T. Lei, and B. Linares-Barranco, "Smart traffic navigation system for fault-tolerant edge computing of internet of vehicle in intelligent transportation gateway," *IEEE transactions on intelligent transportation systems*, 2023.

[29] L. Li, T. Lv, P. Huang, and P. T. Mathiopoulos, "Cost optimization of partial computation offloading and pricing in vehicular networks," *Journal of Signal Processing Systems*, vol. 92, no. 12, pp. 1421–1435, 2020.

[30] Z. Mlika and S. Cherkaoui, "Network slicing with mec and deep reinforcement learning for the internet of vehicles," *IEEE Network*, vol. 35, no. 3, pp. 132–138, 2021.

[31] W. Feng, N. Zhang, S. Li, S. Lin, R. Ning, S. Yang, and Y. Gao, "Latency minimization of reverse offloading in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5343–5357, 2022.

[32] H. Materwala, L. Ismail, and H. S. Hassanein, "Qos-sla-aware adaptive genetic algorithm for multi-request offloading in integrated edge-cloud computing in internet of vehicles," *Vehicular Communications*, vol. 43, p. 100654, 2023.

[33] Y. Ju, Y. Chen, Z. Cao, L. Liu, Q. Pei, M. Xiao, K. Ota, M. Dong, and V. C. Leung, "Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[34] Z. Nan, S. Zhou, Y. Jia, and Z. Niu, "Joint task offloading and resource allocation for vehicular edge computing with result feedback delay," *IEEE Transactions on Wireless Communications*, 2023.

[35] J. Yang, F. Lin, C. Chakraborty, K. Yu, Z. Guo, A.-T. Nguyen, and J. J. Rodrigues, "A parallel intelligence-driven resource scheduling scheme for digital twins-based intelligent vehicular systems," *IEEE Transactions on Intelligent Vehicles*, 2023.

[36] S.-K. Liao, W.-Q. Cai, J. Handsteiner, B. Liu, J. Yin, L. Zhang, D. Rauch, M. Fink, J.-G. Ren, W.-Y. Liu *et al.*, "Satellite-relayed intercontinental quantum network," *Physical review letters*, vol. 120, no. 3, p. 030501, 2018.

[37] Q. Zhang, F. Xu, Y.-A. Chen, C.-Z. Peng, and J.-W. Pan, "Large scale quantum key distribution: challenges and solutions," *Optics express*, vol. 26, no. 18, pp. 24 260–24 273, 2018.

[38] D. Stucki, M. Legre, F. Buntschu, B. Clausen, N. Felber, N. Gisin, L. Henzen, P. Junod, G. Litzistorf, P. Monbaron *et al.*, "Long-term performance of the swissquantum quantum key distribution network in a field environment," *New Journal of Physics*, vol. 13, no. 12, p. 123001, 2011.

[39] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. Dynes *et al.*, "The secoqc quantum key distribution network in vienna," *New Journal of Physics*, vol. 11, no. 7, p. 075001, 2009.

[40] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka *et al.*, "Field test of quantum key distribution in the tokyo qkd network," *Optics express*, vol. 19, no. 11, pp. 10 387–10 409, 2011.

[41] A. Aguado, E. Hugues-Salas, P. A. Haigh, J. Marhuenda, A. B. Price, P. Sibson, J. E. Kennard, C. Erven, J. G. Rarity, M. G. Thompson *et al.*, "Secure nfv orchestration over an sdn-controlled optical network with time-shared quantum key distribution resources," *Journal of Lightwave Technology*, vol. 35, no. 8, pp. 1357–1362, 2017.

[42] Y. Cao, Y. Zhao, Y. Wu, X. Yu, and J. Zhang, "Time-scheduled quantum key distribution (qkd) over wdm networks," *Journal of Lightwave Technology*, vol. 36, no. 16, pp. 3382–3395, 2018.

[43] Y. Cao, Y. Zhao, J. Wang, X. Yu, Z. Ma, and J. Zhang, "Sdqaas: Software defined networking for quantum key distribution as a service," *Optics express*, vol. 27, no. 5, pp. 6892–6909, 2019.

[44] Y. Zuo, Y. Zhao, X. Yu, A. Nag, and J. Zhang, "Reinforcement learning-based resource allocation in quantum key distribution networks," in *Asia Communications and Photonics Conference*. Optica Publishing Group, 2020, pp. T3C–6.

[45] A. Patil, M. Pant, D. Englund, D. Towsley, and S. Guha, "Entanglement generation in a quantum network at distance-independent rate," *npj Quantum Information*, vol. 8, no. 1, p. 51, 2022.

[46] R. Renner, "Security of quantum key distribution," *International Journal of Quantum Information*, vol. 6, no. 01, pp. 1–127, 2008.

[47] 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 Transactions on Wireless Communications*, 2023.

[48] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE communications surveys & tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[49] S. R. White, "Density matrix formulation for quantum renormalization groups," *Physical review letters*, vol. 69, no. 19, p. 2863, 1992.

[50] ——, "Density-matrix algorithms for quantum renormalization groups," *Physical review b*, vol. 48, no. 14, p. 10345, 1993.

[51] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, "Matrix product state representations," *arXiv preprint quant-ph/0608197*, 2006.

[52] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, H.-S. Goan, and Y.-J. Kao, "Variational quantum reinforcement learning via evolutionary optimization," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015025, 2022.

[53] C. Psomas and I. Krikidis, "Wireless powered mobile edge computing: Offloading or local computation?" *IEEE Communications Letters*, vol. 24, no. 11, pp. 2642–2646, 2020.

[54] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *Ieee Access*, vol. 6, pp. 12 825–12 837, 2018.

[55] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[56] J. Barrett, L. Hardy, and A. Kent, "No signaling and quantum key distribution," *Physical review letters*, vol. 95, no. 1, p. 010503, 2005.

[57] M. Cao, H. Wang, T. Yuan, K. Xu, K. Lei, and J. Wang, "Meta-regulation: Adaptive adjustment to block size and creation interval for blockchain systems," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3702–3718, 2022.