Quantum Neural Networks for MADRL-assisted Optimal Resource Allocation in Vehicular Networks

Yuxiang Zheng*, Simon L. Cotton†, Hyundong Shin‡, and Trung Q. Duong*†
*Memorial University, Canada (e-mails: {y.zheng, tduong}@mun.ca)
†Queen's University Belfast, UK (e-mail: {simon.cotton, trung.q.duong}@qub.ac.uk)
‡Kyung Hee University, South Korea (e-mail: hshin@khu.ac.kr)

***Abstract*—In this work, the benefits of employing quantum neural networks (QNNs) in reinforcement learning (RL)-based methods used in vehicular networks are explored. We substitute the classical-bit-based neural networks (NNs) in the multi-agent deep RL (MADRL) with QNNs and propose a QNN-based quantum MADRL (QMADRL) framework to solve a resource allocation (RA) problem in a cellular-vehicle-to-everything (C-V2X) network. The objective of the optimisation is to minimise the age of information (AoI) for vehicle-to-infrastructure (V2I) communications, maximise the delivery probability of the cooperative awareness messages (CAMs) for the vehicle-to-vehicle (V2V) communications, and jointly minimise the power and energy consumption to promote green communication practices. Compared to classical MADRL methods, the proposed QMADRL framework delivers substantially faster convergence while achieving comparable performance after convergence.**

I. Introduction

Vehicle-to-everything (V2X) communications enables vehicles to exchange information about surrounding traffic conditions and potential hazards in complex urban environments. It is expected to play a vital role in future intelligent transportation systems (ITS), particularly for autonomous driving [1]. Vehicle platooning is a promising application of V2X communications for realising autonomous driving, where nearby vehicles travelling in the same lane are grouped into a platoon and orchestrated using vehicle group intelligence (VGI) [2]. Although vehicle platooning can effectively increase the efficiency of traffic control and traffic flow, an optimised resource allocation (RA) method is still required to handle intensive vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, given the stringent reliability and low-latency requirements [3].

To address the optimisation problem, reinforcement learning (RL)-based algorithms have emerged as a promising solution for enhancing reliable and low-latency communications and coordination among vehicles [4]–[8], thus supporting the establishment of platoons with VGI in transportation. In [4], the impact of V2X information on deep RL (DRL)-based platoon controllers is evaluated. A multi-agent RL (MARL)-based mixed vehicle platoon forming method is proposed in [5] for adapting dynamic mixed traffic environments. A MARL and deep Q-learning based algorithm is presented in [6] to optimise the channel and power assignments for vehicle platoons in cellular-V2X (C-V2X) systems. In [7], the age of information (AoI) [3] is considered and jointly optimised with the power consumptions of platoon-leading vehicles (LVs) and the exchange of cooperative awareness messages (CAMs) [9] using a MARL algorithm. In addition to [7], an energy-efficient multi-agent DRL (MADRL) algorithm is proposed in [8], where the AoI, CAM exchange, and joint power-energy allocations are collectively optimised.

TABLE I: Overview of prior research and this work.

Studies	Scope				Method		
	ITS	Vehicle Platooning	AoI	Energy Saving	Classical	HQC	Quantum
[12], [13]	-	-	-	-	-	-	✓
[14]	✓	-	-	-	-	✓	-
[4]–[6]	✓	✓	-	-	✓	-	-
[7]	✓	✓	✓	-	✓	-	-
[8]	✓	✓	✓	✓	✓	-	-
[15]	✓	✓	✓	✓	-	✓	-
This work	✓	✓	✓	✓	-	-	✓

The RL-based algorithms proposed in the aforementioned studies are composed of classical-bit-based neural networks (NNs), whose complexity increases with the number of neurons and layers [10]. The problem of complexity becomes significant when using multiple layers of NNs in DRL to manage complicated systems. To address this issue, quantum neural network (QNN)-based RL (QRL) methods have been proposed, as they offer the potential for improved performance [11] and reduced complexity [12]. In [13], a quantum MARL (QMARL) framework employing variational quantum circuits (VQCs) is proposed in a single-hop offloading environment. It is shown to deliver better performance than both classical and hybrid quantum-classical (HQC) MARL frameworks. A vehicle routing problem is studied in [14] with near-term quantum devices using an HQC heuristic. In [15], an energy-efficient HQC MADRL algorithm, employing VQC, demonstrates remarkable potential for quantum computing in complicated environments. A comparative evaluation with the classical MADRL algorithm proposed in [8] shows that the HQC algorithm effectively optimises the RA problem for vehicle platoons in C-V2X systems. Based on the research findings above, quantum-inspired optimisation algorithms are gaining recognition as promising solutions for complex RA challenges in future networks [16].

The related works are summarised in Table. I, which shows that previous and current studies mainly focus on classical methods for vehicle platooning in ITS, while quantum-based algorithms remain largely unexplored. In particular, QNN-based RL methods have yet to see significant exploration or application. Inspired by recent studies that leverage the principles of quantum mechanics, such as superposition and entanglement, to accelerate training and achieve improved performance [12], [13], [15], we propose a QNN-based quantum MADRL (QMADRL) framework to achieve optimal RA in AoI-aware energy-efficient platoon-based C-V2X networks.

II. System Model and Problem Formulation

A. Platoon-based C-V2X communication network

With $M \in \mathbb{N}^+$ vehicle platoons, each consisting of $V \in \mathbb{N}^+$ vehicles, a single-cell C-V2X system at an intersection is illustrated in Fig. 1. The central single-antenna base station (BS) gathers platoon-state information and disseminates it to all the platoons, thereby ensuring the vehicular system operates efficiently and safely. The leading vehicle in each platoon, the LV, is responsible for V2I communications with the BS to update the platoon-state information and V2V communications with its platoon-member vehicles (MVs) to exchange the CAMs. An indicator $\alpha_{m,t} \in \{0,1\}$ represents the selection of V2I and V2V communications, where t is the time step index. $P_{m,t}$, selected by the m^{th} LV (LV$_m$), denotes the V2I or V2V transmit power, where $m \in \{1,2,\ldots,M\}$. The set of MVs in the m^{th} platoon is denoted as MV$_{m,v}$, where $v \in \{1,2,\ldots,V\}$.

We assume the wireless channel of this system is divided into K orthogonal subchannels, each V2I or V2V link occupies at most one subchannel $k \in \{1,2,\ldots,K\}$, and each subchannel is only available for one link. The channel fading is assumed to be constant at each time step t, length Δt, and independent across all the subchannels. Another indicator $\beta_{m,t} \in \{0,k\}$ denotes the channel assignment. $\beta_{m,t} = k$ means the subchannel k is assigned to LV$_m$, while 0 means no channel is assigned. The signal-to-interference-plus-noise-ratio (SINR) of the V2I and V2V links on subchannel $\beta_{m,t}$, and the corresponding achievable wireless communication rates are written as

$$\text{SINR}^{(\text{vi})}_{m,\beta,t} = \frac{(1-\alpha_{m,t}) f(\beta_{m,t}) P_{m,t} H^{(\text{vi})}_{m,\beta,t}}{\sigma^2 + \sum\limits_{m' \neq m} f(\beta_{m',t}) P_{m',t} H^{(\text{vv}_\text{v})}_{m',\beta,t}}, \qquad \mathcal{C}^{(\text{vi})}_{m,\beta,t} = W \log_2\left(1 + \text{SINR}^{(\text{vi})}_{m,\beta,t}\right), \tag{1}$$

$$\text{SINR}^{(\text{vv})_v}_{m,\beta,t} = \frac{\alpha_{m,t} f(\beta_{m,t}) P_{m,t} H^{(\text{vv})_v}_{m,\beta,t}}{\sigma^2 + \sum\limits_{m' \neq m} f(\beta_{m',t}) P_{m',t} H^{(\text{vx})}_{m',\beta,t}}, \qquad \mathcal{C}^{(\text{vv})_v}_{m,\beta,t} = W \log_2\left(1 + \text{SINR}^{(\text{vv})_v}_{m,\beta,t}\right), \tag{2}$$

where $f(\beta_{m,t}) = (1 - \delta(\beta_{m,t}, 0))$, $f(\beta_{m,t})$ equals 1 when $\beta_{m,t} \neq 0$, and equals 0 when $\beta_{m,t} = 0$. (vi) represents the V2I communications, which is subject to interference arising from the V2V communications, $(\text{vv})_v$ denotes the V2V communications between LV$_m$ and MV$_{m,v}$, and (vx) indicates that the V2V communications can be impacted by interference from both V2I and V2V communications. W is the subchannel bandwidth, and H represents the channel fading power of the wireless link, which is composed of both small-scale and large-scale fading. The fading model considers the Rayleigh fading and log-normal distribution.

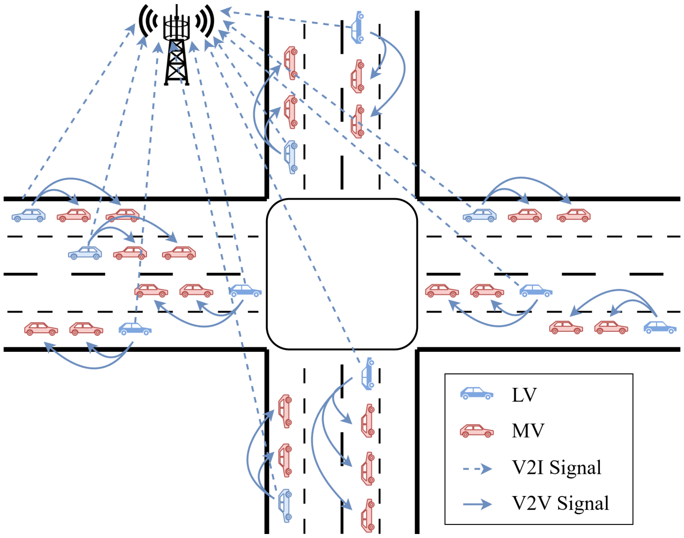


Fig. 1: A C-V2X network at an intersection.

AoI measures how frequently the LVs update their platoon-state information to the BS, which is calculated as the time gap between the newest and previous V2I communications [15]:

$$A_{m,t} = \begin{cases} A_{m,t-1} + \Delta t, & \text{if } \mathcal{C}^{(\text{vi})}_{m,\beta,t} < \mathcal{C}^{(\text{vi})}_{\text{min}}, \\ \Delta t, & \text{otherwise}, \end{cases} \tag{3}$$

which is increased by Δt if the V2I rate is less than the minimum requirement and reset to Δt if the V2I communication succeeds. In addition, the V2V communications are evaluated via the delivery of CAMs with a fixed data size D within a limited time. Based on [9], [17], the CAM delivery interval is set as $T \cdot \Delta t \in [100, 1000]$ ms, where T is the number of overall time steps. A successful CAM delivery is defined as

$$\text{CAM}_m = \left\{ \sum_{t=1}^{T} \sum_{\beta} \min_{(\text{vv})_v} \left\{ \mathcal{C}^{(\text{vv})_v}_{m,\beta,t} \right\} \Delta t \geq D \right\}, \tag{4}$$

which requires all the vehicles in the platoon to finish exchanging CAMs within the time limitation.

The overall optimisation problem for the m^{th} platoon is then formulated as minimising the AoI in (3), maximising the probability of the successful CAM delivery in (4), and realising the joint power-energy allocations as in [8], [15]:

$$\min_{\alpha,\beta,P,E} \left\{ \sum_{t=1}^{T} \left(\frac{A_{m,t}}{T}, \frac{P_{m,t}}{T}, E_{m,t} \right), -\mathbb{P}\Big(\text{CAM}_m\Big) \right\}, \tag{5}$$

$$\text{s.t.} \quad P_{m,t} \in [0, P_{\max}], \forall m, t, \tag{5a}$$

$$\text{if } \beta_{m,t} \neq 0, \beta_{m,t} \neq \beta_{m',t}, \forall m, t, \tag{5b}$$

where $E_{m,t} = P_{m,t} \cdot \Delta t$ is the energy consumed by LV$_m$ at t, $\mathbb{P}(\cdot)$ represents the probability, constraint (5a) limits the power selection, and constraint (5b) guarantees that at most one subchannel is assigned to at most one V2I or V2V link.

B. Multi-agent Deep Deterministic Policy Gradient

A MADRL approach with the deep deterministic policy gradient (DDPG) algorithm is proposed in this section to solve the mixed-integer nonlinear programming problem (5). With each LV_m considered as an agent, the action space and state space at time t for the m^{th} agent, $\mathbf{A}_m$, is defined as

$$\mathcal{A}_{m,t} = \Big[\alpha_{m,t}, \beta_{m,t}, g\left(P_{m,t}, E_{m,t}\right)\Big], \tag{6}$$

$$\mathcal{S}_{m,t} = \begin{bmatrix} \mathrm{SINR}^{(\mathrm{vi})}_{m,\beta,t}, \mathrm{SINR}^{(\mathrm{vv})_v}_{m,\beta,t}, \\ L_{m,t}, \sigma^2, A_{m,t}, D', T' \end{bmatrix}. \tag{7}$$

Four actions are generated in the action space, where the function $g(\cdot)$ decides the final transmit power selection based on the generated $P_{m,t}$ and $E_{m,t}$. $\mathbf{A}_m$ observes the SINR, its location $L_{m,t}$, and the noise power σ^2 from the environment and stores them into the state space together with its AoI level, remaining CAM size D', and remaining time budget T'.

This work considers the decomposed multi-agent DDPG (DE-MADDPG) algorithm proposed in [18], in which a single-agent DDPG algorithm operates locally for each agent and the global critic from the multi-agent DDPG (MADDPG) algorithm coordinates the system-level performance. The corresponding local and global reward functions are designed as

$$\mathcal{R}^G_t = -\frac{1}{M}\sum_{m,\beta}\mathcal{G}\left(f(\beta_{m,t})P_{m,t}H^{(\mathrm{vx})}_{m,\beta,t}\right), \tag{8}$$

$$\begin{aligned}\mathcal{R}^L_{m,t} = & \underbrace{-\mathcal{F}_1\left(P_{m,t}\right) - \mathcal{F}_2\left(\sum_{y=t-\tau}^{t}\rho^{t-y}E_{m,y}\right)}_{\text{Joint power-energy reward}} \\ & \underbrace{-\mathcal{F}_3\left(A_{m,t}\right) + w\cdot\mathbf{1}_{\{\mathcal{C}^{(\mathrm{vi})}_{m,\beta,t} - \mathcal{C}^{(\mathrm{vi})}_{\min}\}}}_{\alpha_{m,t}=0,\ \text{V2I reward}} - \underbrace{\mathcal{F}_4\left(\frac{D'}{D}, \frac{T'}{T}\right)}_{\alpha_{m,t}=1,\ \text{V2V reward}}.\end{aligned} \tag{9}$$

The global reward is calculated as the average interference level adjusted by the function $\mathcal{G}$ among all the assigned channels, with the aim of encouraging agents to select channels that minimise interference for others. The local reward is composed of the joint power-energy reward, the V2I reward, and the V2V reward, where the energy is discounted by the factor ρ, with τ past time steps considered, a gain w is added when a V2I update is successful, and the remaining CAM data and time budget are jointly evaluated. All the items in the local reward function are adjusted to a suitable range via functions $\mathcal{F}_1$–$\mathcal{F}_4$.

Based on the above settings and by utilising the twin delayed deep deterministic policy gradient (TD3) technique [19], the proposed DE-MADDPG algorithm with TD3 for the vehicular network in this work is formulated as

$$\begin{aligned}\underbrace{\nabla_{\theta_m}\mathcal{J}}_{\text{Local actor}} = & \underbrace{\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{B}}\Big[\nabla_{\theta_m}\pi_m\left(a_m|s_m\right)\nabla_{a_m}Q^G_{\psi_1}(\mathbf{s},\mathbf{a})\Big]}_{\text{Global critic}} \\ & \underbrace{+\mathbb{E}_{s_m,a_m\sim\mathcal{B}}\Big[\nabla_{\theta_m}\pi_m\left(a_m|s_m\right)\nabla_{a_m}Q^L_{\phi_m}\left(s_m,a_m\right)\Big]}_{\text{Local critic}},\end{aligned} \tag{10}$$

where $\nabla_{\theta_m}\mathcal{J}$ is the target function at time t for the local actor, $\mathbf{s} = (s_1, ..., s_M)$ and $\mathbf{a} = (a_1, ..., a_M)$ are the states and actions of all the platoons, $\mathcal{B}$ is the experience replay buffer, the first global Q-function $Q^G_{\psi_1}$ from the TD3 technique is chosen, and θ_m, ψ_1, and ϕ_m parameterise the agent policy π_m, $Q^G_{\psi_1}$, and local Q-function $Q^L_{\phi_m}$, respectively. The loss functions for the global and local critics are formulated as

$$\begin{aligned}\mathcal{L}^G(\psi_i) &= \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{r}^G,\mathbf{s}'}\left[\left(Q^G_{\psi_i}(\mathbf{s},\mathbf{a}) - y^G\right)^2\right], i=1,2, \\ y^G &= r^G + \gamma\min_i Q^G_{\psi'_i}\left(\mathbf{s}',\mathbf{a}'\right)\Big|_{a'_m=\pi'_m(s'_m)},\end{aligned} \tag{11}$$

$$\begin{aligned}\mathcal{L}^L(\phi_m) &= \mathbb{E}_{s_m,a_m,r^L_m,s'_m}\left[\left(Q^L_{\phi_m}\left(s_m,a_m\right) - y^L_m\right)^2\right], \\ y^L_m &= r^L_m + \gamma Q^L_{\phi'_m}\left(s'_m,a'_m\right)\Big|_{a'_m=\pi'_m(s'_m)},\end{aligned} \tag{12}$$

where $\mathbf{s}' = (s'_1, ..., s'_M)$ and $\mathbf{a}' = (a'_1, ..., a'_M)$ denote next sets of states and actions, $Q^G_{\psi'_i}$, $Q^L_{\phi'_n}$, and π'_m are the target Q-functions and the target policy. The Q-function at time step t is calculated as the expected value of the discounted return:

$$Q^{\pi_m}(s,a) = \mathbb{E}_{\pi_m}\left[\sum_{y=0}^{\infty}\gamma^y\mathcal{R}^{t+y+1}\Big|s_{m,t},a_{m,t}\right], \tag{13}$$

where γ is the discount factor. Optimisation of the Q-functions is conducted via the QNN proposed in the next section.

III. Quantum Neural Networks

In this section, a detailed description of the QNN proposed for the QMADRL framework is presented. The QNN for all three networks—global critic, local critic, and local actor networks—share a common circuit architecture comprising three primary stages: quantum encoding, quantum processing, and measurement and output mapping. Each stage plays a crucial role in transforming classical RL data into quantum representations, processing these representations through parameterised quantum operations, and finally mapping the quantum measurement results back to classical Q-values.

A. Quantum Encoding

The encoding stage shown in Fig. 2 transforms classical data into quantum states via multiple processes, including feature importance weighting, non-linear transformations, dimensional reduction, and quantum state preparation. For the input vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$, where d is the dimensionality of the input space, our enhanced dimensional reduction process proceeds as follows:

$$\begin{aligned}\mathbf{x}''' &= \tilde{\mathbf{x}} \odot \mathtt{softplus}\left(\mathbf{w}_f\right), \\ \mathbf{x}'' &= \mathtt{tanh}\left(\mathbf{x}''' \odot \mathbf{w_s} + \mathbf{b}_s\right), \\ \mathbf{x}' &= \mathtt{LeakyReLU}\left(\mathbf{x}''\mathbf{W}_1 + \mathbf{b}_1\right), \\ \mathbf{x} &= \mathtt{tanh}\left(\mathbf{h}\mathbf{W}_2 + \mathbf{b}_2\right)\pi.\end{aligned} \tag{14}$$

In this operation sequence, $\odot$ represents element-wise multiplication between vectors. $\mathtt{softplus}(\mathbf{w}_f) = \ln(1 + e^{\mathbf{w}_f})$ is a smooth approximation of the rectified linear unit (ReLU) function that ensures positive weights while maintaining differentiability. The vector $\mathbf{w}_f$ contains trainable feature importance weights that determine the relative contribution of each input feature. Parameters $\mathbf{w}_s$ and $\mathbf{b}_s$ are trainable scaling and shifting parameters that control the non-linear transformation of weighted features. Then the hyperbolic tangent function, denoted as $\mathtt{tanh}$, maps the results after scaling and shifting into the range $[-1, 1]$. The matrices $\mathbf{W}_1 \in \mathbb{R}^{d\times h_d}$ and $\mathbf{W}_2 \in \mathbb{R}^{h_d\times n_q}$, along with bias vectors $\mathbf{b}_1$ and $\mathbf{b}_2$, implement a two-layer transformation where h_d represents the hidden dimension

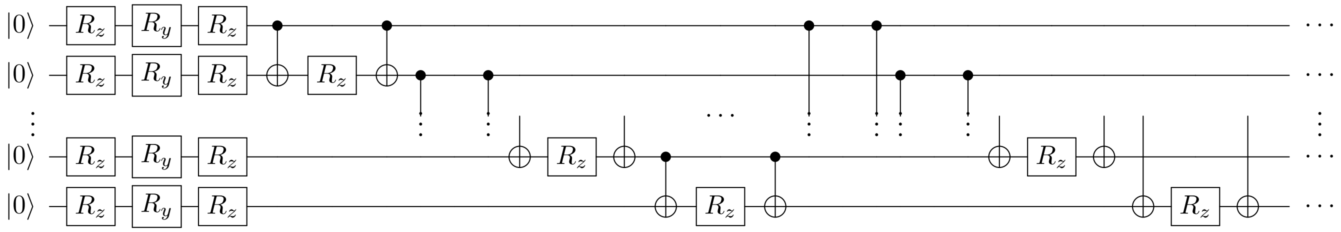


Fig. 2: Quantum encoding stage.

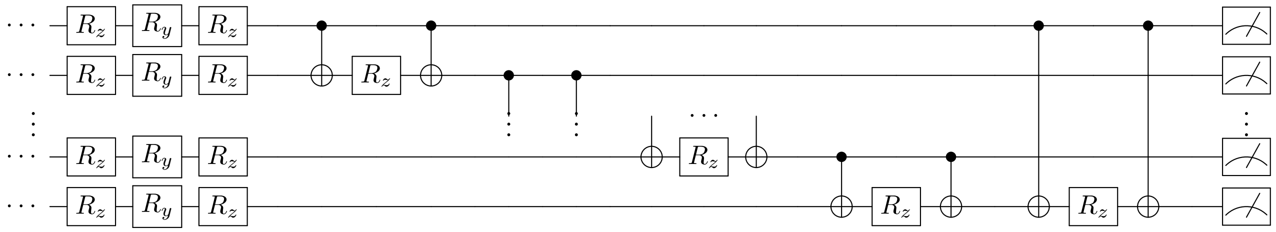


Fig. 3: Quantum processing stage.

and n_q is the number of qubits in the quantum circuit. The activation function $\texttt{LeakyReLU}(x) = \max(0.01x, x)$ allows a small, non-zero gradient when the unit is not active, which helps prevent dead neurons during training. Finally, the $\texttt{tanh}$ function is used again to constrain the values of $\mathbf{x}'$ to the range $[-1, 1]$, with the subsequent multiplication by π scaling these values to the range $[-\pi, \pi]$ which is appropriate for quantum rotation angles. The final output $\mathbf{x}$ is the reduced-dimension representation that will be encoded into the quantum state. The quantum state preparation consists of two components that work together to encode both the individual feature values and their pairwise correlations. Firstly, single-qubit rotations are used to encode individual feature values. Each reduced dimension is encoded into a corresponding qubit using a sequence of rotation gates around the Z and Y axes of the Bloch sphere. The resulting quantum state is expressed as

$$|\psi_0\rangle = \prod_{i=0}^{n_q-1} R_z\left(x_i\theta_i^{z_1}\right) R_y\left(x_i\theta_i^{y}\right) R_z\left(x_i\theta_i^{z_2}\right) |0\rangle_i\,, \qquad (15)$$

where $\theta_i^{z_1}$, θ_i^{y}, $\theta_i^{z_2}$ are trainable rotation parameters, $R_z(\phi)$ and $R_y(\phi)$ represent rotation operators around the Z and Y axes by angle ϕ, respectively. The initial state of the quantum circuit is denoted by $|0\rangle_i$. This sequence of $R_z(\phi)R_y(\phi)R_z(\phi)$ is known to be universal for single-qubit operations, enabling the representation of any possible qubit rotation on the Bloch sphere. After encoding individual features, we introduce quantum correlations between adjacent features through controlled operations. The entangling process is described by

$$|\psi_1\rangle = \prod_{i=0}^{n_q-2} \text{CNOT}_{i,i+1} R_z\left(x_i x_{i+1}\gamma_i\right) \text{CNOT}_{i,i+1} |\psi_0\rangle\,, \quad (16)$$

where γ_i represents a trainable interaction strength parameter. The controlled-NOT (CNOT) gate is a two-qubit operation where the state of the target qubit is flipped conditionally based on the state of the control qubit. This operation generates entanglement between adjacent qubits, capturing correlations between neighbouring features. By combining CNOT gates with parameterised R_z rotations, the circuit can effectively encode nonlinear relationships between input features. This encoding scheme efficiently represents high-dimensional classical data within the exponentially large Hilbert space of the quantum system. To further enhance the expressivity, long-range interactions are introduced, given by

$$|\psi_2\rangle = \prod_{(i,j)\in P} \text{CNOT}_{i,j} R_z\left(x_i x_j \delta_{ij}\right) \text{CNOT}_{i,j} |\psi_1\rangle\,, \qquad (17)$$

where P is a set of selected long-range qubit pairs and δ_{ij} are trainable parameters representing the interaction strengths. These interactions enable the encoding of correlations between non-adjacent features, which is particularly important for capturing complex relationships in high-dimensional data.

B. Quantum Processing

After encoding the classical data into a quantum state, the quantum processing layer applies a series of parameterised gates to transform the encoded quantum state, as shown in Fig. 3. The transformation can be expressed as

$$|\psi_3\rangle = \mathcal{U}_{\text{process}}\left(\Theta\right)|\psi_2\rangle\,, \qquad (18)$$

where $\mathcal{U}_{\text{process}}(\Theta)$ represents the overall unitary transformation with trainable parameters Θ. This unitary transformation is the product of two distinct unitary operations $\mathcal{U}_{\text{process}} = \mathcal{U}_1\left(\Theta_1\right)\mathcal{U}_2\left(\Theta_2\right)$, where $\Theta = \Theta_1 \cup \Theta_2$ represents the full set of trainable parameters composed of the parameters from both operations. The first operation, $\mathcal{U}_1\left(\Theta_1\right)$, applies single-qubit rotations on each qubit, which can be described as

$$\mathcal{U}_1\left(\Theta_1\right) = \prod_{i=0}^{n_q-1} R_z\left(\vartheta_i^{z_1}\right) R_y\left(\vartheta_i^{y}\right) R_z\left(\vartheta_i^{z_2}\right)\,, \qquad (19)$$

where $\Theta_1 = \{\vartheta_i^{z_1}, \vartheta_i^{y}, \vartheta_i^{z_2} \mid i = 0, 1, \ldots, n_q - 1\}$ is the set of trainable parameters associated with the single-qubit rotations. These rotations allow the circuit to perform arbitrary

TABLE II: Simulation parameters.

Environmental Parameters	Symbols	Values
Number of platoons	M	4
Number of vehicles in each platoon	V	4
Vehicle gap within a platoon	-	25 m
Maximum transmit power	P_{max}	30 dBm
Noise power	σ^2	-114 dBm
Carrier frequency	-	2 GHz
Number of resource blocks	K	3
Resource block bandwidth	W	180 kHz
CAM message size	D	4 KB
CAM delivery interval	$T \cdot \Delta t$	100 ms [9], [17]
Large-scale fading update period	$T \cdot \Delta t$	100 ms [17]
Fast fading update period	Δt	1 ms [17]
Number of time steps	T	100
Number of episodes	-	400
Training Parameters	**Symbols**	**Values**
Energy discount factor	ρ	0.5
Reward discount factor	γ	0.99
Number of actions (dimension of $\mathcal{A}_{m,t}$)	n_a	4
Number of qubits	n_q	
- Global critic, local critic, local actor		10, 5, 6
Hidden dimension	h_d	
- Global state and action encoders		76, 16
- Local critic state and action encoders		19, 4
- Local actor state encoders		19
Batch size	-	64
Quantum learning rate	-	0.001
Optimiser	-	Adam

transformations on each qubit independently. Following these rotations, the second operation, $\mathcal{U}_2(\Theta_2)$, applies entangling operations between qubits to enable information exchange and create quantum correlations, which is given by

$$\mathcal{U}_2(\Theta_2) = \prod_{i=0}^{n_q-1} \text{CNOT}_{i,j} R_z(\vartheta_i^{zz}) \, \text{CNOT}_{i,j}, \tag{20}$$

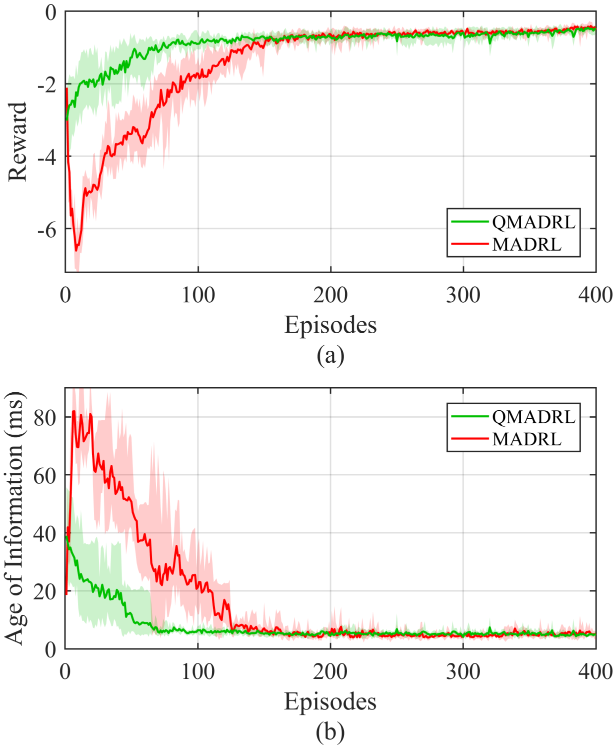
where $\Theta_2 = \{\vartheta_i^{zz} \mid i = 0, 1, \ldots, n_q - 1\}$ is the set of trainable parameters associated with the entangling operations, and the notation $j = (i+1) \bmod n_q$ implements a ring topology which connects the last qubit back to the first one. This structure of rotations and entanglements is inspired by the quantum approximate optimisation algorithm (QAOA), which has been proven effective for optimisation problems on quantum computers. The nearest-neighbour entanglement pattern creates quantum correlations that allow the circuit to represent complex relationships between input features, while maintaining computational efficiency with only $\mathcal{O}(n_q)$ entangling operations instead of the $\mathcal{O}(n_q^2)$ operations required for all-to-all connectivity. This reduces the circuit complexity while preserving the expressivity required for effective quantum processing.

C. Quantum Measurement and Output Mapping

The final stage of the quantum circuit architecture involves measuring the quantum states and mapping the results to classical outputs. For critic networks, a weighted sum of measurements is computed to produce a scalar Q-value:

$$Q = \sum_{i=0}^{n_q-1} w_i \langle \psi_3 | Z_i | \psi_3 \rangle, \tag{21}$$

where Z_i is the Pauli-Z operator applied to the i^{th} qubit, $\langle \psi_3 | Z_i | \psi_3 \rangle$ denotes the expectation value of this measurement of the processed quantum state. The Pauli-Z measurement returns $+1$ for the $|0\rangle$ state and -1 for the $|1\rangle$ state, this yields a real number in the range $[-1, 1]$ that represents the expected outcome of measuring the i^{th} qubit in the Z-basis. Trainable weights w_i determine the contribution of each qubit's measurement to the final Q-value, allowing the network to learn which qubits contain the most relevant information for predicting expected return. For the actor network, the measurements are mapped to a multi-dimensional action vector:



(a)

(b)

Fig. 4: Reward and AoI convergence plots.

$$\mathbf{a} = \texttt{tanh}(\mathbf{Mz} + \mathbf{b}), \tag{22}$$

where $\mathbf{z} \in \mathbb{R}^{n_q}$ contains the expectation values $\langle \psi_3 | Z_i | \psi_3 \rangle$ for each qubit, $\mathbf{M} \in \mathbb{R}^{n_q \times n_a}$ is a trainable output matrix, $\mathbf{b} \in \mathbb{R}^{n_a}$ is a bias vector, and n_a is the action dimension. This process transforms the quantum measurement results into a continuous action vector. The hyperbolic tangent function `tanh` constrains each action component to the range $[-1, 1]$, which is appropriate for normalised action spaces. The trainable output matrix $\mathbf{M}$ and bias vector $\mathbf{b}$ learn the mapping from quantum measurements to optimal actions.

IV. Results and Discussion

The environmental setting in this work follows the 3GPP TR 36.885 urban specification [17], with the key parameters shown in Table II. The simulation is implemented using Python 3.8 and the PyTorch-based TorchQuantum v0.1.8 [20] framework on the NVIDIA RTX A6000 graphics card to accelerate the training of parameterised quantum circuits. The numerical results of our proposed QMADRL framework are compared against the classical simulation in [8] and shown in Fig. 4a and Fig. 4b, with the average taken over five simulation runs.

Fig. 4a illustrates the reward convergence plot from the QMADRL framework. Compared with the classical MADRL method, the quantum algorithm demonstrates faster convergence speed and relatively more stable behaviour. The final system performance of the two approaches after convergence is similar, as the DE-MATD3 method for this system already closely approaches the optimal result obtained through exhaustive search [7]. In Fig. 4b, the variation in the AoI level throughout all the episodes is shown. With its faster convergence speed, as reflected in the reward convergence plot, the AoI from the QMADRL framework also reaches a similar level to the classical method. The average energy consumed by the QMADRL framework in the last 100 episodes is 157.91 mJ, while the classical method consumes 153.35 mJ. This reflects that the energy efficiency characteristic of the energy-focused algorithm in [8] is still well preserved in the quantum design.

Overall, the proposed QMADRL framework achieves a comparable optimal system performance to the classical MADRL method, while exhibiting a significantly faster algorithm convergence speed. Utilising QNNs within the proposed framework holds significant potential for boosting the efficiency of multiple RL agents operating in complex environmental conditions. However, the current design employs relatively simple QNN architectures due to the associated circuit complexity. To further improve the performance of the system, future research should focus on developing more sophisticated QNN models with deeper quantum layers and enhanced parameterisation, while maintaining manageable computational complexity. The advanced architecture could potentially achieve better convergence behaviour, improved generalisation and superior capabilities in large-scale vehicular networks.

V. Conclusion

This work showcases a QNN-based QMADRL framework that solves a mixed-integer nonlinear programming problem for a platoon-based C-V2X network at an intersection. Based on the DE-MADDPG algorithm using the TD3 technique, and quantum circuit design for the QNN, our QMADRL framework reveals a notable performance compared with the classical MADRL approach under the same environmental setting. Numerical results have shown a similar achievable optimal performance as well as a much faster algorithm convergence speed. Future research will focus on enhancing the quantum aspects of the framework. This includes developing deeper and more expressive QNN architectures with optimised parametrisation to improve convergence speed and generalisation capabilities. Moreover, exploring quantum-enhanced models with adaptive circuit depth and efficient resource management will be essential for improving scalability and performance in large-scale vehicular networks. These enhancements will further harness the potential of quantum computing for efficient and scalable resource allocation in next-generation communication systems.

Acknowledgement

The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109 and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941.

References

[1] L. Chen *et al.*, “Milestones in autonomous driving and intelligent vehicles: Survey of surveys,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1046–1056, Feb. 2023.

[2] C. Wu, Z. Cai, Y. He, and X. Lu, “A review of vehicle group intelligence in a connected environment,” *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1865–1889, Jan. 2024.

[3] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Proc. 2011 8th Annu. IEEE Commun. Soc. Conf. Sens. Mesh Ad Hoc Commun. Netw.*, Salt Lake City, UT, USA, Jun. 2011, pp. 350–358.

[4] L. Lei, T. Liu, K. Zheng, and L. Hanzo, “Deep reinforcement learning aided platoon control relying on V2X information,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 5811–5826, Jun. 2022.

[5] Y. Shi, H. Dong, C. R. He, Y. Chen, and Z. Song, “Mixed vehicle platoon forming: A multi-agent reinforcement learning approach,” *IEEE Internet Things J.*, Feb. 2025, DOI: 10.1109/JIOT.2025.3535732.

[6] H. V. Vu *et al.*, “Multi-agent reinforcement learning for channel assignment and power allocation in platoon-based C-V2X systems,” in *Proc. 2022 IEEE 95th Veh. Technol. Conf. VTC2022-Spring*, Helsinki, Finland, Jun. 2022, pp. 1–5.

[7] M. Parvini, M. R. Javan, N. Mokari, B. Abbasi, and E. A. Jorswieck, “AoI-aware resource allocation for platoon-based C-V2X networks via multi-agent multi-task reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 9880–9896, Aug. 2023.

[8] Y. Zheng *et al.*, “Multi-agent DRL for resource allocation in AoI-aware energy-efficient C-V2X networks,” in *Proc. 2024 IEEE 29th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw. (CAMAD)*, Athens, Greece, Oct.21–23 2024, pp. 1–6.

[9] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, ETSI Std. EN 302 637-2, Apr. 2019.

[10] M. Bianchini and F. Scarselli, “On the complexity of neural network classifiers: A comparison between shallow and deep architectures,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1553–1565, Aug. 2014.

[11] A. Abbas *et al.*, “The power of quantum neural networks,” *Nat. Comput. Sci.*, vol. 1, no. 6, p. 403–409, Jun. 2021.

[12] B. Narottama and S. Y. Shin, “Quantum neural networks for resource allocation in wireless communications,” *IEEE Trans. Wirel. Commun.*, vol. 21, no. 2, pp. 1103–1116, Feb. 2022.

[13] W. J. Yun *et al.*, “Quantum multi-agent reinforcement learning via variational quantum circuit design,” in *Proc. 2022 IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Bologna, Italy, Jul. 2022, pp. 1332–1335.

[14] U. Azad, B. K. Behera, E. A. Ahmed, P. K. Panigrahi, and A. Farouk, “Solving vehicle routing problem using quantum approximate optimization algorithm,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7564–7573, Jul. 2023.

[15] Y. Zheng, S. L. Cotton, O. A. Dobre, and T. Q. Duong, “Quantum multi-agent deep reinforcement learning for energy-efficient vehicular networks,” in *Proc. ICC 2025 - IEEE Int. Conf. Commun.*, Montreal, Canada, Jun.8–12 2025, accepted.

[16] T. Q. Duong *et al.*, “Quantum-inspired real-time optimization for 6g networks: Opportunities, challenges, and the road ahead,” *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1347–1359, Aug. 2022.

[17] 3GPP, “Study on LTE-based V2X services,” 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.885, 2016, version 14.0.0.

[18] H. U. Sheikh and L. Bölöni, “Multi-agent reinforcement learning for problems with combined individual and team reward,” in *Proc. Int. Joint Conf. Neural Netw.*, Glasgow, UK, Jul. 2020, pp. 1–8.

[19] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, pp. 1587–1596.

[20] H. Wang *et al.*, “QuantumNAS: Noise-adaptive search for robust quantum circuits,” in *Proc. 2022 IEEE Int. Symp. High-Perform. Comput. Archit. HPCA*, Seoul, Korea, Republic of, Apr. 2022, pp. 692–708.