Quantum DRL for UAV-RIS-aided Maritime Communications with 6G Digital Twin Applications

Sasinda C. Prabhashana*, Dang Van Huynh*, and Trung Q. Duong*†
* Memorial University, Canada, e-mails: {cwelhengodag, vdhuynh, tduong}@mun.ca
† Queen's University Belfast, UK, e-mail: trung.q.duong@qub.ac.uk

***Abstract*—This paper presents a digital twin (DT) framework for the sixth-generation (6G) maritime communication, targeting extreme ultra-reliable low-latency communication (xURLLC). In particular, we propose a DT model, where the physical system integrates unmanned aerial vehicles with reconfigurable intelligent surfaces and a high-altitude platform featuring mobile edge computing. We formulate a nonlinear programming problem to minimize total xURLLC user latency while satisfying energy and computational constraints. Quantum proximal policy optimization (Q-PPO) employed at the DT layer is adopted to cope with dynamic network conditions. The simulation results demonstrate superior performance for Q-PPO, which converges rapidly with minimal state space and requires fewer iterations per episode compared to classical PPO. Hybrid-Q-PPO, incorporating a parameterized quantum circuit within the policy network, also delivers notable performance enhancements than classical PPO. Moreover, Q-PPO achieves higher overall rewards than Hybrid-Q-PPO. Both Q-PPO and Hybrid-Q-PPO significantly reduce latency and improve resource utilization, highlighting the effectiveness of QDRL.**

I. Introduction

The emergence of sixth-generation (6G) wireless communication is set to bring significant advancements in maritime communications, delivering ultra-reliable, low-latency, and high-capacity connectivity to meet the complex demands of modern maritime operations [1]. A key breakthrough in 6G is the evolution of ultra-reliable and low-latency communication (URLLC) into extreme URLLC (xURLLC), which seeks to achieve "nine-nines" reliability (99.9999999%) and sub-millisecond latency [2], [3]. This unprecedented level of reliability and responsiveness is essential for critical maritime applications, including autonomous vessels, real-time navigation, and emergency response systems [4]. Moreover, the integration of artificial intelligence into xURLLC enhances operational efficiency through intelligent decision-making, predictive analytics, and adaptive resource management, ensuring that maritime communication systems remain robust, scalable, and secure [5].

Furthermore, in maritime communications, unmanned aerial vehicles (UAVs) play a vital role in real-time data transmission for applications like vessel tracking, search and rescue and ocean monitoring, while also supporting underwater internet of things by cooperating with unmanned surface and underwater vehicles [6]. Additionally, UAV integration with mobile edge computing (MEC) allows for efficient task offloading, reducing latency and enhancing computational performance [7]. Moreover, UAVs can be equipped with reconfigurable intelligent surfaces (RIS) to enhance communication reliability. UAVs can carry RIS to dynamically adjust phase shifts and redirect signals, ensuring consistent line-of-sight (LoS) connections even in complex maritime environments. This capability not only boosts signal quality but also allows UAVs to extend coverage and overcome obstacles like sea surface reflections and interference, making them an essential component of modern maritime communication networks [8].

In addition, high altitude platforms (HAPs) are also becoming a key asset in enhancing maritime communication networks. While UAVs offer flexibility and mobility, HAPs provide a more stable and long-term solution by operating from the stratosphere, ensuring consistent connectivity [9]. Moreover, HAPs are equipped with advanced computing resources and energy reserves, enabling them to offload computationally intensive tasks from maritime devices, reducing latency and enhancing overall system performance [4]. Furthermore, the concept of digital twin (DT) has gained significant attention in recent years. DT is a virtual representation of physical space that continuously mirrors their real-world counterparts through data integration and dynamic updates to virtual space [7]. This facilitate the seamless synchronization between physical and virtual spaces, allowing for predictive analytics, enhanced decision-making, and optimized operations [10]. With all these resources in the maritime networks, managing them efficiently becomes a significant challenge. This is where deep reinforcement learning (DRL) plays an essential role. DRL enables intelligent, real-time decision-making in maritime communication networks, optimizing resource allocation and managing complex, dynamic environments [8].

The emergence of quantum computing has brought a paradigm shift in various applications within communication systems. Due to its ability to leverage quantum parallelism and superposition, quantum computing allows for the rapid processing of large state and action spaces, which is especially useful in dynamic and complex scenarios [11]. By combining quantum computing with DRL, communication systems can achieve more efficient decision-making processes. For example, quantum circuits can encode classical information into quantum states, enabling the agent to explore vast solution spaces with fewer resources [12]. In large action-space environments, quantum approaches like energy-based models and hybrid algorithms can outperform classical methods by effectively approximating policy functions and action-value functions with fewer parameters [13]. As a result, integrating quantum computing into DRL frameworks not only speeds up learning but also improves the quality of the learned policies in real-time applications [14]. Moreover, challenges remain in efficiently scaling quantum-enhanced DRL algorithms for real-world communication scenarios, such as maritime communications, particularly under the constraints posed by current quantum hardware limitations and noise [11], [13]. Therefore, further investigation is needed to explore new quantum algorithms, hybrid approaches to fully unlock the potential of quantum-enhanced reinforcement learning.

In this paper, we propose an optimization framework for joint task offloading and bandwidth allocation in a MEC-aided maritime communication network. The optimization problem is formulated as a nonlinear programming problem that seeks to minimize total user latency while adhering to energy consumption and computational resource constraints. To efficiently solve this problem, we employ quantum proximal policy optimization (Q-PPO) algorithm, enabling the framework to adapt to varying network conditions and user demands. The simulation results demonstrate promising performance of Q-PPO and Hybrid-Q-PPO over classical PPO algorithm.

II. System Model

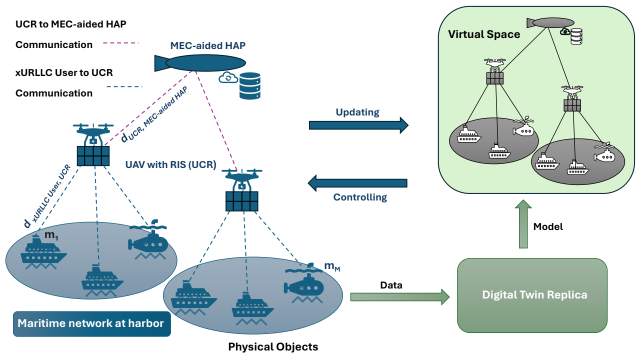

Fig. 1: An illustration of digital twin-enabled 6G maritime communication air-ground integrated network architecture.

Fig. 1 illustrates the DT-enabled maritime communication network at a harbor, which is designed to meet the stringent latency requirements of xURLLC users by integrating a terrestrial communication framework with aerial components. This physical objects are mapped to a DT model to replicate the physical objects and operations in the virtual space, enabling continuous monitoring, updating, and controlling of the system. The xURLLC users, denoted by the set $\mathcal{M} = \{1, \dots, m, \dots, M\}$, are connected to HAP that manages their communication needs. To address the challenges posed by non-line-of-sight (NLoS) communication, these xURLLC users leverage UAVs, represented by the set $\mathcal{U} = \{1, \dots, u, \dots, U\}$. Each UAV is equipped with a passive RIS containing a set of passive reflecting elements N. These elements, carried by the UAV, are responsible for reflecting signals, thereby creating a LoS path toward the HAP, overcoming NLoS limitations. The signal reflection process is mathematically represented by the diagonal matrix $\mathbf{\Phi}$, where $\mathbf{\Phi} = \text{diag}(e^{j\phi_1}, e^{j\phi_2}, \dots, e^{j\phi_N})$. This matrix describes the phase shifts applied by each reflecting element in the RIS, which are essential for steering the reflected signals toward the HAP. The HAP is equipped with a set of K antennas and incorporates a mobile edge computing (MEC) node, which enhances the provision of edge computing services to the xURLLC users, further reducing latency. The system is modeled using a three-dimensional (3D) Cartesian coordinate system. xURLLC users are positioned on the horizontal plane, with their height set to zero and the coordinates are set to $l_m(t) = \{x_m(t), y_m(t), 0\}$, while the UAVs carrying the RIS are assumed to operate at a fixed altitude R. The coordinate of the u-th UAV is given by $l_u(t) = \{x_u(t), y_u(t), R\}$. Moreover, HAP is also located at a fixed altitude $Y \geq R$ and the coordinates is given by $l_H(t) = \{x_H(t), y_H(t), Y\}$. In this paper, we considered UAVs are already deployed and clustered using the K-means clustering algorithm [15].

A. Channel Modeling

1) xURLLC users to HAP via UAV-Carried RIS (UCR): In this study, we model the channel vector between the m-th user and the UAV-Carried RIS (UCR) as $\mathbf{h}_{m,u}(t) \in \mathbb{C}^{N\times 1}$. To analyze the channel behavior, we utilize the Rician fading model combined with free-space path loss. Given the dynamic nature of the UCR, the effects of the non-line-of-sight (NLoS) components are considered negligible, simplifying the expression for the channel gain vectors. Therefore, at time t, the channel vector $\mathbf{h}_{m,u}(t)$ can be expressed as [8]:

$$\mathbf{h}_{m,u}(t) = \sqrt{\left(\frac{4\pi f_c d_{m,u}(t)}{c}\right)^{-\alpha}} \left(\sqrt{\frac{\Psi}{\Psi+1}}\mathbf{h}^{LoS}_{m,u}(t)\right), \quad (1)$$

where f_c is the carrier frequency of the transmission signal, and c denotes the speed of light. The path loss exponent is represented by α, and Ψ is the Rician factor. The Euclidean distance between the m-th user and the u-th UCR is given by $d_{m,u}(t) = \sqrt{(x_u(t) - x_m(t))^2 + (y_u(t) - y_m(t))^2 + R^2}$. Furthermore, at time t, the line-of-sight (LoS) component $\mathbf{h}^{LoS}_{m,u}(t) \in \mathbb{C}^{N\times 1}$ is calculated as $\mathbf{h}^{LoS}_{m,u}(t) = \left[1, e^{-j\frac{2\pi}{\lambda}d_u \cos(\phi_{\text{AoA}}(t))}, \dots, e^{-j\frac{2\pi}{\lambda}(N-1)d_u \cos(\phi_{\text{AoA}}(t))}\right]^T$, where λ is the wavelength of the transmission signal, d_u is the uniform spacing between the RIS elements, and $\phi_{\text{AoA}}(t)$ is the angle of arrival (AoA) [8].

2) UCR to HAP: Similarly, the channel matrix between the UCR and the HAP, $\mathbf{H}_{u,H}(t) \in \mathbb{C}^{K\times N}$, can be formulated as:

$$\mathbf{H}_{u,H}(t) = \sqrt{\left(\frac{4\pi f_c d_{u,H}(t)}{c}\right)^{-\alpha}} \left(\sqrt{\frac{\Psi}{\Psi+1}}\mathbf{H}^{LoS}_{u,H}(t)\right), \quad (2)$$

where $d_{u,H}(t)$ is the Euclidean distance between the UCR and the HAP, which can be calculated as $d_{u,H}(t) = \sqrt{(x_u(t) - x_H(t))^2 + (y_u(t) - y_H(t))^2 + (R-Y)^2}$. Moreover, $\mathbf{H}^{LoS}_{u,H}(t) \in \mathbb{C}^{K\times N}$ is given by $\mathbf{H}^{LoS}_{u,H}(t) = \mathbf{a}_H(\phi_{AoA}(t))\mathbf{a}^H_u(\phi_{AoD}(t))$. The steering vector for the HAP, $\mathbf{a}_H(\phi_{AoA}(t)) \in \mathbb{C}^{K\times 1}$, is calculated as: $\mathbf{a}_H(\phi_{AoA}(t)) = \left[1, e^{-j\frac{2\pi}{\lambda}d_H \cos(\phi_{AoA}(t))}, \dots, e^{-j\frac{2\pi}{\lambda}(K-1)d_H \cos(\phi_{AoA}(t))}\right]^T$, where d_H is the spacing between the uniform linear array HAP antennas. Similarly, the steering vector for the UCR, $\mathbf{a}_u(\phi_{AoD}(t)) \in \mathbb{C}^{N\times 1}$, represents the phase shifts introduced by the N elements of the RIS as the signal is reflected towards the HAP. It is expressed as: $\mathbf{a}_u(\phi_{AoD}(t)) = \left[1, e^{-j\frac{2\pi}{\lambda}d_u\cos(\phi_{AoD}(t))}, \dots, e^{-j\frac{2\pi}{\lambda}(N-1)d_u\cos(\phi_{AoD}(t))}\right]^T$ [8]. Therefore, cascaded channel between the user m and the HAP can be formulated as $\mathbf{g}_{m,H}(t) = \mathbf{H}_{u,H}(t)\mathbf{\Phi}_u(t)\mathbf{h}_{m,u}(t)$.

B. Communication Model

xURLLC users are enabled to offload their computationally heavy tasks to the HAP through UCR. At the HAP, the instantaneous signal-to-interference-plus-noise ratio (SINR) of m-th user can be expressed as follows [8]:

$$\gamma_m^H(t) = \frac{p_m|\mathbf{g}_{m,H}(t)|^2}{\sum_{j=1,j\neq m}^{M} p_j|\mathbf{g}_{j,H}(t)|^2 + z^2(t)}, \tag{3}$$

where p_m represents the transmit power of user m, and z_t is the instantaneous noise power, characterized by a Gaussian complex normal distribution $\sim \mathcal{CN}\left(0,\sigma^2\right)$. Therefore, the achievable data rate of the m-th user can be calculated as [7]

$$R_m(t) \approx b_m(t)B\left(\log_2\left(1+\gamma_m^H(t)\right) - \sqrt{\frac{V_m(t)}{N}}\frac{Q^{-1}(\epsilon)}{\ln 2}\right). \tag{4}$$

Where $b_m(t) \in [0,1]$ is the allocated bandwidth for user m and B represents the system bandwidth. Moreover, N is the block length. The parameter ϵ corresponds to the decoding error probability. The function $Q^{-1}(\cdot)$ refers to the inverse of the Q-function, which is defined as $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(-\frac{t^2}{2}\right)dt$. $V_m(t)$ represents the channel dispersion, which is given by the expression $V_m(t) = 1 - \left[1+\gamma_m^H(t)\right]^{-2}$.

C. Task offloading Modeling

1) Latency Model: A task from the m-th user is represented as a 3-tuple $x_m = \{s_m, c_m, L_m^{\max}\}$, where s_m denotes the task size in bits, c_m represents the computational requirement in CPU cycles, and $L_m^{\max}$ is the maximum allowable latency for task completion. The total processing latency for the m-th xURLLC user depends on whether the task is processed locally or offloaded to the MEC server. If processed locally, the processing latency L_m^{local} can be calculated as $L_m^{\text{local}} = \frac{c_m}{f_m^{\text{local}}(t)}$, where $f_m^{\text{local}}(t)$ is the local CPU processing speed in CPU cycles per second. In contrast, if the task is offloaded, the transmission latency L_m^{tx} is calculated as $L_m^{\text{tx}} = \frac{s_m}{R_m(t)}$. Moreover, the processing delay at the MEC server, denoted as L_m^{pr}, is given by $L_m^{\text{pr}} = \frac{c_m}{f_m^{\text{mec}}(t)}$, where $f_m^{\text{mec}}(t)$ is the processing speed at the MEC server in CPU cycles per second. If user m offloads the task, the offloading latency L_m^{offload} can be calculated as:

$$L_m^{\text{offload}} = L_m^{\text{pr}} + L_m^{\text{tx}}, \tag{5}$$

Therefore, the total latency for the m-th user can be expressed as:

$$L_m^{\text{total}} = (1-y_m(t))L_m^{\text{local}} + y_m(t)L_m^{\text{offload}}, \tag{6}$$

where $y_m(t) \in [0,1]$ represents the offloading fraction, determining the portion of the task processed locally versus offloaded to the MEC.

2) Energy Model: The energy consumption for each xURLLC user m is determined by whether it processes the task locally or offloads it to the MEC. If the device processes the task locally, the energy consumed is given by $E_m^{\text{local}}(t) = \kappa_m c_m (f_m^{\text{local}}(t))^2$, where κ_m is the energy efficiency coefficient of the m-th device processor [7]. On the other hand, if the task is offloaded, the energy consumed for offloading can be calculated as:

$$E_m^{\text{offload}}(t) = p_m L_m^{\text{tx}}. \tag{7}$$

Therefore, the total energy consumption for device m can be formulated as:

$$E_m^{\text{total}}(t) = (1-y_m(t))E_m^{\text{local}}(t) + y_m(t)E_m^{\text{offload}}(t). \tag{8}$$

D. Problem Formulation

In this paper, our goal is to minimize the total latency experienced by each xURLLC user m. To achieve this, we define the objective function that quantifies the total latency for each xURLLC user m.

$$\Omega(y(t), b(t)) = \sum_{m=1}^{M}(1-y_m(t))L_m^{\text{local}} + y_m(t)\left(L_m^{\text{tx}} + L_m^{\text{pr}}\right), \tag{9}$$

Then, the optimization problem is formulated as follows.

$$\begin{aligned}
\textit{\textbf{(P1):}} \quad & \min_{y(t),b(t)} \quad \Omega(y(t), b(t)), && \text{(10a)}\\
\text{s.t.} \quad & L_m^{\text{total}} \le L_m^{\max}, \quad \forall m \in \mathcal{M}, && \text{(10b)}\\
& E_m^{\text{total}}(t) \le E_m^{\max}, \quad \forall m \in \mathcal{M}, && \text{(10c)}\\
& \sum_{m=1}^{M} y_m(t) f_m^{\text{mec}}(t) \le F^{\text{mec}}, \quad \forall m \in \mathcal{M}, && \text{(10d)}\\
& (1-y_m(t))f_m^{\text{local}}(t) \le F^{\text{local}}, \quad \forall m \in \mathcal{M}, && \text{(10e)}\\
& 0 \le y_m(t) \le 1, \quad \forall m \in \mathcal{M}, && \text{(10f)}\\
& 0 \le b_m(t) \le 1, \sum_{m=1}^{M} b_m(t) \le 1 \quad \forall m \in \mathcal{M}, && \text{(10g)}
\end{aligned}$$

As specified in (10), constraint (10b) ensures that the total latency for each user m does not exceed the maximum allowable latency. Similarly, constraint (10c) restricts the total energy consumption for each user m to stay within the maximum allowable limit. Constraint (10d) indicates that the total computation resources allocated at the MEC server cannot exceed its maximum CPU frequency. Additionally, constraint (10e) ensures that sufficient computational resources are available locally to process the task. Constraint (10f) requires the offloading fraction $y(t)$ to remain between 0 and 1 for all $m \in \mathcal{M}$. Moreover, constraint (10g) ensures that the bandwidth allocation coefficient $b(t)$ for each user is also between 0 and 1 and total bandwidth allocation does not exceed 1 for all $m \in \mathcal{M}$.

III. Quantum-Proximal Policy Optimization Based Solution

The problem formulated in (10a) is computationally challenging. To address this complexity and pave the way for the application of quantum DRL, we propose Q-PPO algorithm due to its well-established robustness and stability in training reinforcement learning agents, especially in high-dimensional and continuous action spaces which align with the complexities of our maritime xURLLC scenario.

A. MDP Formulation

To utilize the Q-PPO framework for solving (10), the problem must first be formulated as a Markov decision process, defined by the 3-tuple $\mathcal{S}, \mathcal{A}, \mathcal{R}$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, and $\mathcal{R}$ specifies the reward function.

1) State space: The state space $s(t)$ at time t can be defined as $s(t) = \{\gamma_m(t), L_m^{\text{total}}, L_m^{\text{offload}}\}$ where $\gamma_m^{HAP}(t)$ is the SINR value for each xURLLC user at time t. L_m^{total} and L_m^{offload} denote the total delay and offloading time for each xURLLC user at time t respectively.

2) Action space: The action space is represented as $a(t) = \{b(t), y(t)\}$ where $b(t)$ and $y(t)$ are the bandwidth allocation and offloading fraction for each xURLLC user at time t respectively.

3) Reward: The reward function can be formulated as

$$r(t) = \frac{1}{W}\Bigg(-\sum_{m=1}^{M} L_m^{\text{total}} - \lambda_1 f_{c1}(t) - \lambda_2 f_{c2}(t) \\ - \lambda_3 f_{c3}(t) - \lambda_4 f_{c4}(t)\Bigg), \qquad (11)$$

where $\lambda_1 f_{c1}(t), \lambda_2 f_{c2}(t), \lambda_3 f_{c3}(t)$ and, $\lambda_4 f_{c4}(t)$ are the penalty functions for the satisfaction of the constraints (10b),(10c),(10d), and (10e) and λ_i is the penalty factor for each constraint with i=1,2,3,4. Moreover, W is the scaling factor. The f_{c1} is defined as the $f_{c1} = \sum_{m=1}^{M} \max\left(0, L_m^{\text{total}} - L_m^{\text{max}}\right)$. This formulation ensures that no penalty is added when the constraint is satisfied. Conversely, when a constraint is violated, when $L_m^{\text{total}} > L_m^{\text{max}}$ max function returns a positive value that scales with the extent of the violation. Similarly, $f_{c2}(t) = \sum_{m=1}^{M} \max\left(0, E_m^{\text{total}} - E_m^{\text{max}}\right), f_{c3}(t) = \max\left(0, \sum y_m(t) f_m^{\text{mec}}(t) - F^{\text{mec}}\right)$, and $f_{c4}(t) = \max\left(0, \sum (1 - y_m(t)) f_m^{\text{local}}(t) - F^{\text{local}}\right)$ are modeled.

B. Encoding Classical States to Quantum States

Based on the formulated state space, the classical state data, $s(t)$, is transformed into quantum state data using a higher-order encoding, also known as a feature map. In Qiskit, this encoding is implemented using the ZZFeatureMap [16]. The operation of ZZFeatureMap can be denoted as U_x^{encode}, which maps the classical state $s(t)$ to the encoded quantum state $|s(t)\rangle$, represented as $U_x^{\text{encode}} : s(t) \mapsto |s(t)\rangle$. Here, $|s(t)\rangle$ is the quantum state encoding the classical features in Hilbert space H and serves as the input for the parametric quantum circuits (PQCs) in the Q-PPO framework.

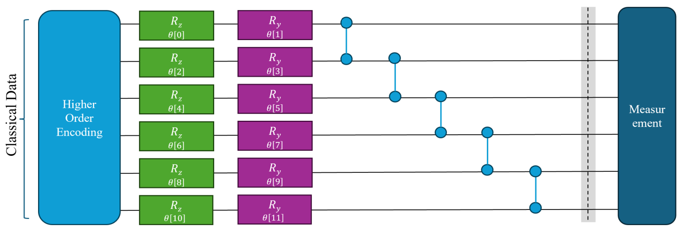

Fig. 2: Proposed PQC for policy and value networks of the Q-PPO framework

C. Parametric Quantum Circuit (PQC) for Q-PPO

The policy network employs a PQC, denoted as $U_{\text{policy}}(\theta)$, to model the policy function $\pi_\theta(a(t) \mid s(t))$, where θ represents the learnable parameters, $s(t)$ is the current state, and $a(t)$ is the action. The PQC is constructed using the unitary operation:

$$U_{\text{policy}}(\theta) = \prod_{l=1}^{L_{\text{policy}}} \left[\left(\prod_{j=0}^{n-2} \text{CZ}_{j,j+1} \right) \times \left(\prod_{i=0}^{n-1} R_y^{(i)}\left(\theta_{l,i,y}\right) R_z^{(i)}\left(\theta_{l,i,z}\right) \right) \right], \qquad (12)$$

where L_{policy} is the number of layers, $R_z^{(i)}\left(\theta_{l,i,z}\right)$ is a parameterized rotation around the Z-axis applied to qubit i, and $R_y^{(i)}\left(\theta_{l,i,y}\right)$ is a parameterized rotation around the Y-axis applied to qubit i. The $\text{CZ}_{j,j+1}$ gates create entanglement between adjacent qubits j and $j+1$. Moreover, n is the number of qubits. This PQC transforms the input quantum state through these sequential gate operations, resulting in an output state that represents the policy.

Similarly, the value network uses a separate PQC, $U_{\text{value}}(\phi)$, with parameters ϕ, to estimate the value function $V_\phi(s(t))$, which indicates the expected cumulative reward for the given state. The unitary operation for the value network follows a similar form:

$$U_{\text{value}}(\phi) = \prod_{l=1}^{L_{\text{value}}} \left[\left(\prod_{j=0}^{n-2} \text{CZ}_{j,j+1} \right) \times \left(\prod_{i=0}^{n-1} R_y^{(i)}\left(\phi_{l,i,y}\right) R_z^{(i)}\left(\phi_{l,i,z}\right) \right) \right], \qquad (13)$$

where L_{value} denotes the number of layers for the value network. This structure enables the value network to approximate the value function by applying quantum operations that encode information about the state. To optimize these PQC networks, the parameters θ and ϕ are iteratively updated by minimizing their respective loss functions. The policy network's loss function is based on the policy advantage, while the value network's loss function minimizes the mean squared error between the predicted and target values. Gradient descent with the parameter-shift rule is used to refine these parameters progressively, enhancing the performance of the PQCs within the Q-PPO framework. Moreover, Hybrid-QPPO is designed by integrating the PQC with the policy network.

D. Quantum Measurement

In the Q-PPO framework, the quantum measurement process is essential for obtaining meaningful outputs from the PQCs of both the policy and value networks. For the policy network, the measurement involves computing the expectation values of a set of observables, denoted as O_i, where each observable corresponds to a different action dimension i. These observables are typically constructed as Pauli strings, such as $O_i = Z_i \otimes I \otimes \cdots \otimes I$, where the Pauli-$Z$ operator acts on the i-th qubit. The expectation value for each observable, $\langle O_i \rangle$, is computed directly using the Estimator primitive [16], which provides exact expectation values assuming infinite shots. These expectation values, inherently in the range $[-1, 1]$, are mapped to $[0, 1]$ using the transformation $(\langle O_i \rangle + 1)/2$, yielding valid action probabilities for the policy network.

Similarly, for the value network, the measurement aims to estimate the value function $V_\phi(s(t))$. A single observable, $O = Z \otimes I \otimes \cdots \otimes I$, is used. The expectation value $\langle O \rangle$ is calculated directly using the Estimator, serving as the estimated value function for the given state $s(t)$. For both networks, the expectation values are computed based on the final quantum state obtained from the unitary operations $U_{\text{policy}}(\theta)$ and $U_{\text{value}}(\phi)$, respectively. This quantum measurement process integrates the outputs of the PQCs into the Q-PPO framework, guiding action selection and state evaluation to improve the agent's performance. The detailed algorithm for the proposed Q-PPO framework is described in Algorithm 1.

Algorithm 1 Proposed Quantum-Proximal Policy Optimization (Q-PPO)

```
1:  Initialize environment, quantum policy network $U_{\text{policy}}(\theta)$,
    value network $U_{\text{value}}(\phi)$, encoding circuit $U_x^{\text{encode}}$, and
    hyperparameters $\alpha_{\text{policy}}, \alpha_{\text{value}}, \gamma, \lambda, \epsilon, K, M, T$.
2:  for episode = 1 to $M$ do
3:      Reset environment; observe $s_0$.
4:      Collect batch $\mathcal{D}$ by interacting with the environment:
5:      for time step $t = 1$ to $T$ do
6:          Encode state $s(t)$ into quantum state $|s(t)\rangle$ using $U_x^{\text{encode}}$.
7:          Compute $\pi_\theta(a \mid s(t))$ by applying $U_{\text{policy}}(\theta)$ to $|s(t)\rangle$.
8:          Sample action $a(t)$ from $\pi_\theta(a \mid s(t))$.
9:          Execute $a(t)$; observe $r(t)$, $s(t+1)$.
10:         Store $\{s(t), a(t), r(t), s(t+1), \pi_\theta(a(t) \mid s(t))\}$ in $\mathcal{D}$.
11:     end for
12:     Compute value estimates $V_\phi(s(t))$ by applying $U_{\text{value}}(\phi)$
        to $|s(t)\rangle$.
13:     Compute advantages $A(t)$ using GAE:
        $$\delta_t = r(t)+\gamma V_\phi(s(t+1))-V_\phi(s(t)), \quad A(t) = \sum_{l=0}^{T-t}(\gamma\lambda)^l \delta_{t+l}$$
14:     Set $V_{\text{target}}(t) = A(t) + V_\phi(s(t))$.
15:     Set $\theta_{\text{old}} \leftarrow \theta$.
16:     for epoch $k = 1$ to $K$ do
17:         Shuffle $\mathcal{D}$ and create mini-batches.
18:         for each mini-batch do
19:             Compute $r_t(\theta) = \frac{\pi_\theta(a(t) \mid s(t))}{\pi_{\theta_{\text{old}}}(a(t) \mid s(t))}$.
20:             Compute policy loss:
                $$L_{\text{policy}} = -\frac{1}{|\text{batch}|}\sum_{t\in\text{batch}}\Big[\min\big(r_t(\theta)A(t), \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)A(t)\big)\Big]$$
21:             Update $\theta$ using gradient descent:
                $$\theta \leftarrow \theta - \alpha_{\text{policy}}\nabla_\theta L_{\text{policy}}$$
22:             Compute value loss:
                $$L_{\text{value}} = \frac{1}{|\text{batch}|}\sum_{t\in\text{batch}}\left(V_\phi(s(t)) - V_{\text{target}}(t)\right)^2$$
23:             Update $\phi$ using gradient descent:
                $$\phi \leftarrow \phi - \alpha_{\text{value}}\nabla_\phi L_{\text{value}}$$
24:         end for
25:     end for
26: end for
```

IV. Numerical Results And Discussions

A. Simulation Settings

The simulation parameters are set as follows: the policy and value learning rates are 5×10^{-5} and 4×10^{-4}, respectively. The discount factor γ is 0.99, with a GAE parameter λ of 0.95. We use a clipping parameter ϵ of 0.1, applying 10 epochs per update. Training occurs over 1000 episodes, each capped at a maximum of 20 steps with a batch size of 16. The PQC consists of 2 quantum layers. The system setup includes 3 xURLLC users. RIS consists of 8 reflective elements and 4 antennas at the HAP. The system bandwidth is set at 30 MHz, with a Rician factor of $K = 12$, noise power is -110 dBm/Hz, and a path loss exponent $\alpha = 2$. Penalty factors are defined as $\lambda_1 = 1$, $\lambda_2 = 0.5$, $\lambda_3 = 1 \times 10^{-8}$, and $\lambda_4 = 1 \times 10^{-8}$. Task sizes for users range from 5×10^5 to 6×10^5 bits, with task complexities p_m ranging between 1×10^8 and 1.2×10^8 cycles. F^{mec} is 10 GHz and F^{local} is set at 2 GHz. Maximum energy consumption per xURLLC user is $E_m^{\max} = 0.030$ J and a latency limit of $L_m^{\max} = 0.020$ s. Users are located at $(0, 0, 0)$, $(0, 100, 0)$, and $(50, 0, 0)$, with the UAV positioned at $(100, 100, 100)$ and the HAP at $(150, 150, 200)$. We used the IBM Qiskit to develop and execute Q-PPO and Hybrid Q-PPO algorithms.

B. Numerical Results

1) Training performance analysis: As shown in Fig. 3, Q-PPO demonstrates promising performance, achieving higher rewards compared to Hybrid-Q-PPO and classical PPO across training episodes. Q-PPO reaches a stable reward level around episode 500, maintaining consistently higher rewards than both Hybrid-Q-PPO and PPO. Hybrid-Q-PPO, incorporating a PQC in the policy network, also surpasses classical PPO in reward levels, although it remains lower than Q-PPO. Notably, PPO exhibits slower convergence and remains at a relatively low

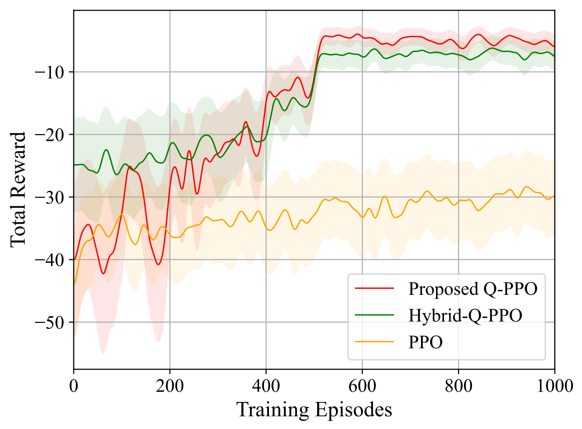

Fig. 3: Training performance: (Each training episode consists 20 iterations in all 3 algorithms: Q-PPO, Hybrid-Q-PPO and PPO)

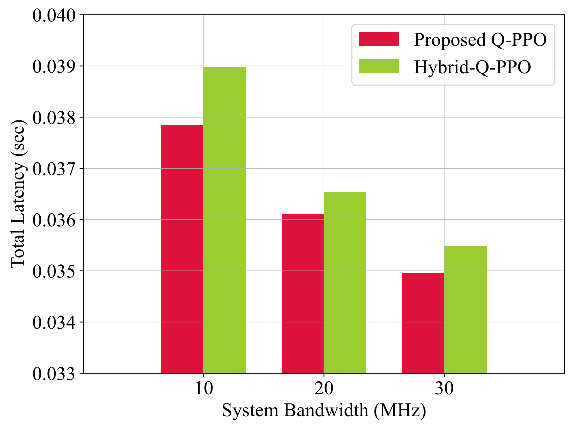

Fig. 4: Total latency with different system bandwidth

reward level.

2) Total latency with different system bandwidth: According to Fig.4, total latency of the system for both Q-PPO and Hybrid-Q-PPO methods decreases as the system bandwidth increases. Comparatively, the Q-PPO achieves lower latency than Hybrid-Q-PPO across all system bandwidths, demonstrating its effectiveness in minimizing total latency. The difference in latency between the two methods is more noticeable at lower bandwidths and gradually narrows as the bandwidth increases, suggesting that Q-PPO provides greater performance benefits under constrained bandwidth conditions.

V. Conclusions

We have investigated a optimization framework for task offloading and bandwidth allocation in MEC-aided maritime communication network targeting 6G xURLLC requirements with a DT approach. By employing Q-PPO and Hybrid-Q-PPO at the DT layer, we have demonstrated that quantum DRL techniques can significantly outperform classical approaches like PPO. Notably, both Q-PPO and Hybrid-Q-PPO achieve faster convergence, reduced state space dependency, and fewer iterations per episode, making them highly efficient for resource-constrained environments. This research highlights quantum DRL's promising role in optimizing complex wireless communication tasks, providing a foundation for further paving the way for efficient, reliable 6G xURLLC communications.

References

[1] F. S. Alqurashi, A. Trichili, N. Saeed, B. S. Ooi, and M.-S. Alouini, "Maritime communications: A survey on enabling technologies, opportunities, and challenges," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3525–3541, Feb. 2023.

[2] C. She, C. Pan, T. Q. Duong, T. Q. S. Quek, R. Schober, M. Simsek, and P. Zhu, "Guest editorial xURLLC in 6G: Next generation ultra-reliable and low-latency communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 7, pp. 1963–1968, Jul. 2023.

[3] J. Park, S. Samarakoon, H. Shiri, M. K. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, "Extreme URLLC: Vision, challenges, and key enablers," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 63–69, Dec. 2020.

[4] N. Nomikos, P. K. Gkonis, P. S. Bithas, and P. Trakadas, "A survey on UAV-aided maritime communications: Deployment considerations, applications, and future challenges," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 56–71, Jan. 2023.

[5] X. Su, L. Meng, and J. Huang, "Intelligent maritime networking with edge services and computing capability," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 606–13 619, Nov. 2020.

[6] Y. Liu, C.-X. Wang, H. Chang, Y. He, and J. Bian, "A novel non-stationary 6G UAV channel model for maritime communications," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 2992–3003, Oct. 2021.

[7] D. V. Huynh, S. R. Khosravirad, A. Masaracchia, O. A. Dobre, and T. Q. Duong, "Edge intelligence-based ultra-reliable and low-latency communications for digital twin-enabled metaverse," *IEEE Commun. Lett.*, vol. 11, no. 8, pp. 1733–1737, Aug. 2022.

[8] H. Yang, S. Liu, L. Xiao, Y. Zhang, Z. Xiong, and W. Zhuang, "Learning-based reliable and secure transmission for UAV-RIS-assisted communication systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 6954–6965, Jul. 2024.

[9] T. T. Bui, A. Masaracchia, V. Sharma, O. Dobre, and T. Q. Duong, "Impact of 6G space-air-ground integrated networks on hard-to-reach areas: Tourism, agriculture, education, and indigenous communities," *EAI Endorsed Transactions on Tourism, Technology and Intelligence*, vol. 1, no. 1, pp. 1–8, Sep. 2024.

[10] D. V. Huynh, V.-D. Nguyen, S. R. Khosravirad, G. K. Karagiannidis, and T. Q. Duong, "Distributed communication and computation resource management for digital twin-aided edge computing with short-packet communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3008 – 3021, Oct. 2023.

[11] F. Zaman, A. Farooq, M. A. Ullah, H. Jung, H. Shin, and M. Z. Win, "Quantum machine intelligence for 6G URLLC," *IEEE Wireless Commun.*, vol. 30, no. 2, pp. 22–29, Apr. 2023.

[12] Silvirianti, B. Narottama, and S. Y. Shin, "Layerwise quantum deep reinforcement learning for joint optimization of UAV trajectory and resource allocation," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 430–441, Jan. 2024.

[13] J. A. Ansere, E. Gyamfi, V. Sharma, H. Shin, O. A. Dobre, and T. Q. Duong, "Quantum deep reinforcement learning for dynamic resource allocation in mobile edge computing-based IoT systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 6221–6232, Jun. 2024.

[14] B. Narottama and S. Aïssa, "Quantum machine learning for performance optimization of RIS-assisted communications: Framework design and application to energy efficiency maximization of systems with RSMA," *IEEE Trans. Wireless Commun.*, vol. 23, no. 5, pp. 1103–1115, May 2024.

[15] Q. Huang, W. Wang, W. Lu, N. Zhao, A. Nallanathan, and X. Wang, "Resource allocation for multi-cluster NOMA-UAV networks," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8448–8459, Nov. 2022.

[16] A. J. Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, , and J. M. Gambetta, "Quantum computing with Qiskit," *Quantum computing with Qiskit. arxiv.org/abs/2405.08810*, May 2024.