Quantum Deep Reinforcement Learning for URLLC Satellite-Air-Ground Integrated Networks with Digital Twin Applications

Sasinda C. Prabhashana, *Student Member, IEEE,* Dang Van Huynh, *Member, IEEE,* Haejoon Jung, *Senior Member, IEEE,* Berk Canberk, *Senior Member, IEEE,* Simon L. Cotton, *Fellow, IEEE,* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—In this paper, we explore a maritime 6G-enhanced satellite-air-ground integrated network (SAGIN) that incorporates a UAV-carried reconfigurable intelligent surface (UCR) relay, and low Earth orbit (LEO) satellites equipped with mobile edge computing (MEC) facilities. The system captures dynamic maritime conditions, including ultra-reliable low-latency communication (URLLC) user mobility and UCR movements across harbor environments. The primary objective is to minimize the total system cost by jointly optimizing task offloading decisions, bandwidth allocation, local computational resource distribution, transmission power control, and caching management, while satisfying strict latency and resource constraints. To address this, we formulate a mixed-integer nonlinear programming (MINLP) problem that captures the complexity of resource optimization in the maritime 6G-enhanced SAGIN. Two quantum-enhanced deep reinforcement learning algorithms, namely quantum-enhanced deep deterministic policy gradient (QEDDPG) and quantum-enhanced proximal policy optimization (QEPPO), are proposed to solve the formulated MINLP problem. Moreover, higher-order quantum feature encoding and quantum neural networks are utilized to accelerate learning and enhance decision-making. Simulation results demonstrate that QEDDPG and QEPPO significantly outperform conventional deep reinforcement learning methods by achieving lower system costs and more efficient resource allocation. These findings shows that the potential of quantum-driven reinforcement learning for enabling scalable, efficient, and intelligent resource management in future 6G-enhanced SAGINs.**

***Index Terms*—6G networks, quantum deep reinforcement learning, ultra-reliable low-latency communications, space-air-ground integrated networks, maritime communications, digital twins, satellite communications, mobile edge computing.**

S. C. Prabhashana and D. V. Huynh are with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, Canada (e-mails: {cwelhengodag, vdhuynh}@mun.ca).

D. V. Huynh is also with the Faculty of Computer Networks and Communications, University of Information Technology, Vietnam National University, Ho Chi Minh City, Quarter 34, Linh Xuan Ward, Ho Chi Minh City, Vietnam (e-mail: danghv@uit.edu.vn).

H. Jung is with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: haejoonjung@khu.ac.kr).

B. Canberk is with the School of Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, U.K., (e-mail: b.canberk@napier.ac.uk).

S. L. Cotton is with the Centre for Wireless Innovation (CWI), School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT3 9DT, Belfast, U.K. (email: simon.cotton@qub.ac.uk).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and with Queen's University Belfast, U.K., and also with Kyung Hee University, South Korea (e-mail: tduong@mun.ca).

This paper was accepted in part for presentation at the IEEE International Conference on Communications (ICC) Workshop, June 2025, Montreal, Canada.

The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, and in part by the NSERC CREATE program (Grant number 596205-2025). The work of H. Jung was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2021-II212046) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). The work of B. Canberk is supported in part by The Scientific and Technological Research Council of Turkey (TUBITAK) Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab Project 5239903. The work of S. L. Cotton was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) through the EPSRC Hub on All Spectrum Connectivity under Grant EP/X040569/1 and Grant EP/Y037197/1.

Corresponding authors are Trung Q. Duong and Haejoon Jung.

I. Introduction

Futuristic maritime operations demand more intelligent communication systems to effectively manage the task automation. Especially these maritime environments are highly dynamic and it is essential to have rapid machine-to-machine communication. These operations demand high data rates, greater efficiency, and stable connectivity to maintain smooth, delay-free performance. [1], [2]. To support such demands, the integration of sixth-generation (6G) wireless networks is essential. 6G networks can bring significant advancements in maritime communications such as delivering ultra-reliable, low-latency, and high-capacity connectivity [3]. Especially, these maritime networks are required to support for mission-critical applications such as autonomous vessel control, real-time situational awareness, and emergency response coordination and task automation [4]–[6]. Integrating ultra-reliable and low-latency communication (URLLC) in to the network can maintain the performance in time-sensitive and mission critical operations [4], [5]. Moreover, managing large-scale operations within highly dynamic maritime environments presents considerable challenges. To address these complexities, the integration of artificial intelligence (AI) serves as a critical enabler for enhancing maritime task automation and communication efficiency [1]. AI technologies are capable of managing multiple operational tasks effectively, while also improving system

performance through intelligent decision-making, predictive analytics, and adaptive resource allocation. This ensures that maritime communication networks remain robust, scalable, and secure in the face of increasing operational demands [7].

Moreover, satellite-air-ground integrated networks (SAGINs) are a promising direction for 6G. Especially, maritime regions face limited communication resources, while SAGIN combines terrestrial, aerial, and satellite infrastructures into a unified system. This integration significantly expands communication coverage. Furthermore, it enables seamless connectivity across vast and remote areas where traditional terrestrial networks are limited or unavailable [8]. SAGINs are particularly crucial in maritime environments, where vessels often operate far from shore-based infrastructure. Therefore, SAGINs use satellite links for wide-area coverage. UAVs for flexible relay and surveillance and ground stations for high-speed backhaul. This combination enables resilient and continuous communication links. [9]. Moreover, this architecture ensures uninterrupted data exchange, enhances situational awareness, and supports real-time operations across distributed maritime users.

In maritime SAGINs, maintaining reliable connectivity is challenging due to user mobility, long transmission distances, and frequent signal blockage near ports. A key component in SAGIN's air layer is the UAV [10]. UAVs play a vital role in maritime communications by enabling real-time data transmission for vessel tracking, search and rescue, and ocean monitoring [11]. They also support the underwater Internet of Things by collaborating with unmanned surface and underwater vehicles [12]. Furthermore, UAVs operate at high altitudes. This offers reliable line-of-sight (LoS) links. These links are especially useful in harbor areas where infrastructure often causes signal blockage [13]. Moreover, their mobility allows dynamic repositioning to maintain connectivity and it acts as aerial relays in complex maritime environments. However, the LoS link may still degrade in complex maritime environments. To overcome this, reconfigurable intelligent surfaces (RISs) have emerged as a promising solution for improving link reliability and spectral efficiency. RIS technology consists of intelligent, controllable surfaces that adjust signal phase and reflection to enhance communication range, reliability, and spectral efficiency [14]. However, most existing RIS installations are fixed to static locations such as walls. This fixed positioning creates limitations when environmental obstacles block signal paths, resulting in degraded system performance. RIS integration with UAVs introduces a mobile solution to overcome these limitations. UAV-carried RIS (UCR) systems combine flight mobility with real-time signal control, enabling dynamic LoS restoration in obstructed areas [15]. Thus, this integration enhances coverage, reduces interference, and enhances spectral utilization in dense deployment scenarios [16].

In maritime communication, maintaining continuous coverage over wide ocean areas is difficult due to terrestrial base stations have limited range. Consequently, satellites play a vital role in ensuring global connectivity within SAGINs [8]. The rapid development and large-scale deployment of satellite constellations have proven essential for delivering reliable, wide-area access services, particularly in geographically isolated, oceanic, and underserved regions [11]. Integrating mobile edge computing (MEC) with satellite networks introduces a transformative shift in distributed processing and content delivery. MEC enables satellites to offer localized communication, computation, and caching capabilities, effectively bringing edge computing services closer to end users [17]. This proximity allows for reduced backhaul dependency and improved response time, especially for latency-sensitive applications in remote maritime zones, offshore platforms [18].

Thus, managing SAGIN resources while maintaining the required quality of service and reliability remains a major challenge. To address this, deep reinforcement learning (DRL) offers a powerful solutions by enabling adaptive, real-time optimization across highly dynamic and complex network environments [19], [20]. Unlike traditional optimization methods, DRL combines deep neural networks with reinforcement learning principles, enabling intelligent agents to make real-time decisions by continuously interacting with the environment and learning optimal policies without requiring prior system knowledge [16]. 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 [21]. This facilitate the seamless synchronization between physical and virtual spaces. In addition, this allows predictive analytics, enhanced decision-making, and optimized operations in networks. [22].

Importantly, the emergence of quantum computing (QC) has introduced a transformative capability in the design and optimization of modern communication systems [23]. Rooted in the principles of quantum mechanics, QC enables the simultaneous processing of multiple states and actions. This makes it particularly well-suited for solving complex problems in dynamic and uncertain environments, where classical approaches often face scalability limitations [24]. When integrated with DRL, QC enhances the ability of communication systems to make faster and more efficient decisions. Quantum circuits allow for the encoding of classical data into quantum states, enabling agents to explore exponentially larger solution spaces while consuming fewer computational resources [25]. As a result, the integration of quantum computing into DRL not only accelerates policy learning but also enhances decision quality in real-time applications, such as resource allocation in 6G-enhanced SAGINs [26]. However, existing studies often rely on early stage quantum models, and the scalability of quantum-enhanced DRL remains constrained by current hardware limitations, including noise sensitivity and qubit coherence. Therefore, further investigation is needed to explore novel quantum algorithms and hybrid reinforcement learning strategies that can adapt to complex, real-world communication environments. Addressing this gap is essential to fully realize the benefits of quantum-enhanced learning frameworks and unlock their full potential in future communication systems.

A. Related Works

Efficient and reliable maritime communications networks are essential in the smooth operation of harbors. The rapid growth of maritime activities demands computation-intensive and latency-sensitive services [1], [11], [18], [27], [28]. In [18], a double-edge computation offloading framework was proposed to support secure and low-latency communication in integrated space-air-aqua networks. In this framework, the UAVs relay data between maritime users, base stations, and satellites. Similarly, a joint trajectory and communication optimization strategy based on multi-agent reinforcement learning was applied in [11]. This approach enhanced the efficiency and fault-tolerance of heterogeneous vehicle networks in maritime search and rescue operations. Furthermore, in [28], a hybrid offshore and aerial-based multi-access computation offloading scheme was proposed. The model leveraged edge servers mounted on both offshore stations and UAVs to minimize latency in marine communication networks. In addition, multivessel computation offloading strategies were introduced to reduce energy consumption and service delay for maritime MEC networks in [27]. Furthermore, DRL-based latency minimization techniques were introduced to optimize UAV-enabled maritime edge networks. These methods combined virtual machine multiplexing with trajectory control in order to improve performance [1]. These advances collectively highlight the importance of integrating MEC, UAV-assisted relaying, and dynamic optimization strategies to support the evolving communication and computational needs in maritime environments. However, most of these studies are based on a static network architecture and do not consider the effect of user mobility and dynamic link variation within the network. Thus, this paper extends prior research by developing a DT-enhanced SAGIN model that captures user mobility and environmental dynamics.

Furthermore, SAGINs have emerged as a key architecture for next-generation wireless systems. These networks offer seamless coverage and integrated connectivity across multiple domains. However, the dynamic topology and inherent heterogeneity of SAGIN create major challenges for efficient resource management and real-time service delivery [8], [29]–[36]. To address these challenges, a generative adversarial network-powered DRL-based routing method was proposed in [29]. This approach improves load balancing in SAGIN by predicting network states and reducing the overhead of continuous information updates. Similarly, a cross-domain virtual network embedding strategy was introduced to orchestrate heterogeneous resources across satellite, aerial, and terrestrial layers in [30]. Notably, a distributed DRL-assisted resource allocation framework was proposed in [31]. It focused on optimizing edge caching and managing high-volume data traffic within SAGIN environments. Moreover, a blockchain-based federated reinforcement learning strategy was proposed to secure traffic offloading processes and protect against malicious attacks in SAGIN systems in [32]. Despite these developments, managing latency under high mobility conditions remains a major concern. To tackle this, a delay-sensitive task offloading and dynamic traffic optimization framework was introduced in [33]. Additionally, a collaborative computing and communication integration strategy for SAGIN was emphasized in [34]. This approach aims to support real-time and computationally intensive services. Specifically, a cost-and delay-constrained anycasting framework was proposed in [35]. It enables flexible resource orchestration between edge and cloud nodes within SAGIN. Thus, the above studies discussed the use of DRL algorithms in various SAGIN architectures. However, this work enhances the learning ability of DRL agents by integrating quantum computing, making the agent capable of exploring a larger and more complex state–action space within the SAGIN environment.

The use of quantum machine learning (QML) in wireless networks has gained increasing attention as a promising tool to enhance computational speed and scalability [25], [37], [38]. Accordingly, a layerwise quantum DRL framework was introduced in in [25]. The proposed algorithm improve UAV trajectory planning and power allocation while enhancing energy efficiency. Furthermore, in [37], a quantum multi agent actor critic network was proposed to facilitate cooperative mobile access among multiple UAVs by addressing scalability limitations with quantum centralized critics networks. Moreover, an adaptive quantum federated learning strategy was proposed in [38]. It improves the surveillance in autonomous drone networks while reducing communication overhead using quantum neural networks (QNN). Additionally, a joint quantum reinforcement learning and neural Myerson auction framework was proposed to enhance digital twin services across multi-tier wireless networks in [39]. Accordingly, QML is used in SAGINs for efficient resource allocation, cooperative scheduling, and enhanced decision-making under dynamic network conditions [8], [23], [40]. In [8], a quantum multi-agent reinforcement learning scheduler was proposed to optimize cooperation between CubeSats and high-altitude long-endurance UAVs in SAGIN. This framework reduce action dimensionality and improving global access services. Similarly, quantum-inspired optimization methods were introduced to accelerate real-time decision-making for resource management across space, air, and ground nodes [23]. Moreover, QNNs with parallel training were proposed to optimize wireless resource allocation under distributed and large-dimensional settings in [40]. This method offers scalability advantages that are critical for SAGIN operations. Furthermore, QNN-based reinforcement learning techniques have demonstrated faster convergence and better handling of dynamic user grouping and transmission resource allocation challenges, This further enhance the adaptability of network architectures [25], [41], [42]. However, the above studies show that combining QNNs with DRL improves performance. Most of them use simple quantum network architectures and basic embedding techniques. In contrast, this work applies an advanced quantum embedding method and a novel QNN architecture. These improvements enhance the QDRL agent's exploration and expressibility, leading to efficient resource allocation in the proposed 6G-

enhanced SAGIN.

B. Motivation and Contributions

In maritime-enhanced SAGINs, efficient and reliable communication is vital for supporting computation-intensive and latency-sensitive applications. However, dynamic topology, heterogeneous components, and harsh marine conditions create challenges for low-latency, energy-efficient service delivery. Conventional DRL methods often struggle with scalability and efficient performance in such dynamic, large-scale networks. These challenges intensify in maritime-enhanced SAGINs, where rapid decision-making and adaptive resource management are crucial. Integrating quantum computing into DRL offers a promising solution, leveraging quantum superposition and entanglement to accelerate policy learning and enhance decision-making. By incorporating quantum feature encoding and QNNs, quantum-enhanced DRL can more effectively manage the vast state-action spaces and resource allocation complexities typical of maritime 6G-enhanced SAGINs.

Inspired by advancements in maritime 6G communications and quantum-enhanced DRL, our study focuses on efficient resource optimization within maritime-enhanced SAGIN. We address this challenge by developing a realistic system model that captures dynamic URLLC user mobility across harbor operations, UAV-aided RIS-based aerial relaying for enhanced signal propagation, satellite-based MEC processing for distributed computing, and intelligent access path switching between terrestrial and space layers. We formulate the joint resource allocation and task offloading problem as a mixed-integer nonlinear programming (MINLP) problem with the aim of minimizing the system cost, subject to constraints on latency, computation resources, caching capacity, bandwidth, and transmission power. To solve this problem, we employ a quantum-enhanced DRL framework to optimize resource management under the complex and dynamic conditions of the 6G-enhanced maritime SAGIN. According to our knowledge, our proposed framework uniquely employs quantum-enhanced DRL for optimizing resource allocation to minimize system costs within a maritime-enhanced SAGIN. The main contributions of our paper can be summarized as follows:

- We develop a realistic maritime-enhanced SAGIN system model in which LEO satellites are responsible for navigation-related tasks, while the BS handles ground operational tasks within the harbor. This model captures critical elements such as dynamic URLLC user mobility across harbor operations, UCR-based aerial relaying for enhanced signal propagation, satellite and terrestrial-based MEC processing for distributed computing, and intelligent access-path switching between the ground and space layers to ensure robust and low-latency maritime communication.
- We formulate the task offloading and resource allocation problem as a MINLP problem aimed at minimizing the system cost, subject to latency, computation, caching, bandwidth, and transmission power constraints. To enable efficient real-time decision-making, we further transform this optimization into a quantum-enhanced Markov decision process (MDP) framework, which is solved using quantum-enhanced DRL techniques.
- We propose two quantum-enhanced DRL algorithms, namely quantum-enhanced deep deterministic policy gradient (QEDDPG) and quantum-enhanced proximal policy optimization (QEPPO), to solve the reformulated MDP. These algorithms leverage higher-order quantum feature encoding and parameterized quantum circuits to accelerate policy learning and enhance decision-making.
- We conduct extensive simulations using realistic maritime traffic patterns and communication models to assess the performance of the proposed quantum-enhanced DRL frameworks. The evaluation results reveal that our approach consistently outperforms conventional DRL algorithms in minimizing system costs and optimizing resource allocation. These findings validate the effectiveness of quantum-enhanced learning for complex decision-making in maritime 6G-enabled SAGINs.

C. Paper Structure and Notations

The structure of this paper is organized as follows. Section II describes the system model and the problem formulation, including URLLC user mobility, UCR mobility, channel modelling, data processing model, and the formulation of the MINLP-based optimization problem. Section III presents the proposed quantum-enhanced DRL solutions designed for the maritime 6G-enhanced SAGIN system. Section IV presents the simulation results along with detailed analysis to evaluate the effectiveness of the proposed approach. Section V concludes the paper by summarizing the key findings and future research directions.

Notations: In this paper, lowercase letters represent scalar values, bold lowercase letters represent vectors, and bold uppercase letters represent matrices. The notation $\mathbf{x} \sim \mathcal{CN}(\mu, \sigma^2)$ means that $\mathbf{x}$ follows a complex Gaussian distribution with mean μ and variance σ^2, while $\mathbf{x} \sim \mathcal{N}(\mu, \sigma^2)$ refers to a real-valued Gaussian distribution. The length of a vector $\mathbf{x}$ is denoted by $|\mathbf{x}|$, and the set of complex numbers is written as $\mathbb{C}$. The variable $x_{m,r}(t)$ refers to the value linked to the m-th transmitter and r-th receiver at time slot t. The function $\text{diag}(\cdot)$ creates a diagonal matrix from a vector. The symbols $(\cdot)^T$ and $(\cdot)^H$ refer to the transpose and the complex-conjugate transpose, respectively. The tensor product between quantum states is shown by $\otimes$. These notations are used consistently throughout the paper for clarity. Furthermore, the summary of key notations is provided in Table I.

II. System Model

In this paper, we propose a digital twin-enhanced SAGIN architecture for maritime communication, as illustrated in Fig. 1. The ground layer consists of M maritime URLLC users, denoted by $\mathcal{M} = \{1, \ldots, M\}$, who dynamically enter and exit the harbor. These users maintain continuous connectivity with

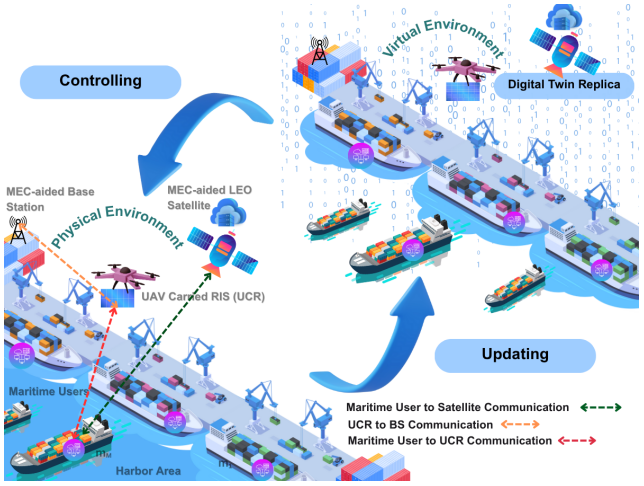

Fig. 1. An illustration of digital twin-enhanced satellite-air-ground integrated maritime communication network architecture.

a LEO satellite, which is equipped with Y antennas, MEC capabilities, and caching resources to support data processing and reduce backhaul load. While the satellite provides broad coverage, meeting the strict latency and reliability requirements of URLLC users necessitates more responsive and localized handling. To address this, a UAV carried passive RIS (UCR) is deployed to serve as an aerial relay. The RIS comprises N reflecting elements that form a controllable LoS path by redirecting signals toward a BS. The reflection is governed by a diagonal phase shift matrix, $\mathbf{\Theta}_u(t) = \text{diag}(e^{j\Theta_1}, e^{j\Theta_2}, \ldots, e^{j\Theta_N})$, enabling effective beam steering. When the UCR is within range, user signals are redirected to the BS, facilitating low-latency, high-reliability transmission. When UCR is out of range, users fall back to direct satellite communication. This flexible switching between access paths ensures robust connectivity under dynamic maritime conditions.

The system is modeled using a three-dimensional Cartesian coordinate system. The BS antennas are positioned at an altitude of H and it's coordinates are given by $\mathbf{q}_{bs} = (x_{bs}, y_{bs}, H) \in \mathbb{R}^3$. Similarly, the LEO satellite's altitude is r, and coordinates are $\mathbf{q}_{sat}(t) = (x_{sat}(t), y_{sat}(t), r) \in \mathbb{R}^3$. With recent advancements in satellite communication and the increased deployment of satellites in low Earth orbit, we assume that LEO satellite coverage is consistently available in the maritime environment.This continuous coverage is assumed through a constellation-based system such as Starlink, where overlapping satellite footprints provide seamless connectivity over coastal and open-sea regions.

A. URLLC User Mobility

The mobility of the users is modeled to reflect realistic behavior in a dynamic maritime environment. We consider the entire harbour operation time into T discrete time slots each of length Δt, which can be denoted as $\mathcal{T} = \{1, 2, \cdots, T\}$. Moreover, we consider that the harbor consists of B berths. Each berth $b \in \mathcal{B} = \{1, \ldots, B\}$ has fixed coordinates $\mathbf{q}_b^{\text{berth}} = (x_b^{\text{berth}}, y_b^{\text{berth}}, 0) \in \mathbb{R}^3$. All users are positioned on the horizontal plane, with their height set to zero, and their coordinates are given by $\mathbf{q}_m(t) = (x_m(t), y_m(t), 0) \in \mathbb{R}^3$ at time t. We define a state variable $S_m(t) \in \{\mathbb{M}, \mathbb{S}, \mathbb{W}\}$ to characterize the behavior of each user over time. Specifically, the state $\mathbb{M}$ represents movement along either an entry or exit route, while the state $\mathbb{S}$ indicates that the user is docked at a designated berth. The state $\mathbb{W}$ corresponds to a waiting condition outside the harbor area. Each user is assigned a predefined entry route composed of ξ_m^{in} waypoints, denoted as $\{w_m^{\text{in},k}\}_{k=1}^{\xi_m^{\text{in}}}$, where each waypoint is given by $w_m^{\text{in},k} = (x_k, y_k)$. The final waypoint of this entry route corresponds to the user's assigned berth, such that $w_m^{\text{in},\xi_m^{\text{in}}} = \mathbf{q}_{b_m(t)}^{\text{berth}}$. Similarly, the exit route consists of ξ_m^{out} waypoints, denoted by $\{w_m^{\text{out},k}\}_{k=1}^{\xi_m^{\text{out}}}$. The first waypoint of the exit route is set at the berth location, defined as $w_m^{\text{out},1} = \mathbf{q}_{b_m(t)}^{\text{berth}}$. Furthermore, berth assignment is dynamically determined by the mapping $b_m(t) \in \mathcal{B} \cup \{0\}$, where $b_m(t) = 0$ indicates that the user is not currently occupying any berth. The occupancy status of each berth is represented by the variable $O_b(t)$, which can be defined as

$$O_b(t) = \begin{cases} m, & \text{if user } m \text{ occupies berth } b, \\ 0, & \text{if berth } b \text{ is unoccupied.} \end{cases} \tag{1}$$

When a user is in the moving state $\mathbb{M}$, the heading angle $\psi_m(t)$ is directed toward the next waypoint, either $w_m^{\text{in},k+1}$ or $w_m^{\text{out},k+1}$. The index k is incremented once the user reaches the current waypoint. Then, the heading angle $\psi_m(t)$ can be calculated as $\psi_m(t) = \arctan2(y_{k+1} - y_m(t-1), x_{k+1} - x_m(t-1))$, where (x_{k+1}, y_{k+1}) are the coordinates of the next waypoint. Moreover, the user's speed $v_m(t)$ can be modeled as

$$v_m(t) = \begin{cases} v_m^{\text{in}} e^{-\kappa ||\mathbf{q}_m(t) - \mathbf{q}_{b_m(t)}^{\text{berth}}||}, & \text{if } S_m(t) = \mathbb{M} \text{ (entering)}, \\ v_m^{\text{out}}, & \text{if } S_m(t) = \mathbb{M} \text{ (exiting)}, \\ 0, & \text{if } S_m(t) \in \{\mathbb{S}, \mathbb{W}\}, \end{cases} \tag{2}$$

where $\kappa > 0$ is the decay constant that governs the speed reduction as users approach their assigned berth during entry. In state $\mathbb{M}$, the position can be calculated as [23]

$$x_m(t) = x_m(t-1) + v_m(t)\cos(\psi_m(t))\Delta t + \Delta x_m(t), \tag{3}$$

$$y_m(t) = y_m(t-1) + v_m(t)\sin(\psi_m(t))\Delta t + \Delta y_m(t), \tag{4}$$

where $\Delta x_m(t), \Delta y_m(t) \sim \mathcal{N}(0, \sigma_m^2)$ are environmental deviations. In state $\mathbb{S}$, $\mathbf{q}_m(t) = \mathbf{q}_{b_m(t)}^{\text{berth}}$. In state $\mathbb{W}$, $\mathbf{q}_m(t) = [x_m^{\mathbb{W}}, y_m^{\mathbb{W}}, 0]$, with $v_m(t) = 0$ and $\Delta x_m(t) = \Delta y_m(t) = 0$. Consequently, user's state transitions occur as follows: from $\mathbb{W}$ to $\mathbb{M}$ when a berth is free ($O_b(t) = 0$). From $\mathbb{M}$ to $\mathbb{S}$ when $\|\mathbf{q}_m(t) - \mathbf{q}_{b_m(t)}^{\text{berth}}\| < \epsilon$. From $\mathbb{S}$ to $\mathbb{M}$ after berth operations and from $\mathbb{M}$ to $\mathbb{W}$ when $\|\mathbf{q}_m(t) - w_m^{\text{out},\xi_m^{\text{out}}}\| < \epsilon$. Where ϵ is the threshold distance.

B. UCR Mobility

The UCR operates at a fixed altitude L and flies continuously over the harbor operation time. Its coordinates at time t are given by $\mathbf{q}_u(t) = (x_u(t), y_u(t), L) \in \mathbb{R}^3$. The UCR follows a planned trajectory consisting of a sequence of ξ_u pre-defined waypoints, expressed as $\{w_u^k\}_{k=1}^{\xi_u}$. Each waypoint, $w_u^k = (x_u^k, y_u^k)$, defines a target position in the horizontal plane. The direction of the UCR's movement, denoted by the angle $\rho_u(t)$, is determined by the vector pointing from its current position to the next waypoint. The position of the UCR at each time slot can be calculated as [43]

$$x_u(t) = x_u(t-1) + v_u \cos(\rho_u(t))\Delta t + \Delta x_u(t), \tag{5}$$

$$y_u(t) = y_u(t-1) + v_u \sin(\rho_u(t))\Delta t + \Delta y_u(t), \tag{6}$$

where v_u is the constant speed of the UCR. Moreover, $\Delta x_u(t)$ and $\Delta y_u(t) \sim \mathcal{N}(0, \sigma_u^2)$ are random disturbances due to environmental factors.

C. Channel Modeling

1) Channel between URLLC users and UCR: In this work, we consider the channel vector between the m-th user and the UCR, denoted by $\mathbf{h}_{m,u}(t) \in \mathbb{C}^{N\times 1}$. We employ a Rician fading model combined with free-space path loss. We assume that the NLoS components are negligible, simplifying the channel representation. At time t, the channel vector can be expressed as [14]

$$\mathbf{h}_{m,u}(t) = \sqrt{\left(\frac{4\pi f_c d_{m,u}(t)}{c}\right)^{-\Omega}} \left(\sqrt{\frac{\gamma}{\gamma+1}}\,\mathbf{h}_{m,u}^{LoS}(t)\right), \tag{7}$$

where f_c denotes the carrier frequency, c represents the speed of light, Ω specifies the path loss exponent, and γ refers to the Rician factor, quantifying the ratio of LoS to scattered signal power. The Euclidean distance between the m-th user and the UCR is expressed as $d_{m,u}(t) = ((x_u(t) - x_m(t))^2 + (y_u(t) - y_m(t))^2 + L^2)^{1/2}$. In this work, we consider a uniform linear array configuration for the RIS elements. Accordingly, LoS component of the channel vector $\mathbf{h}_{m,u}^{\text{LoS}}(t) \in \mathbb{C}^{N\times 1}$ can be derived as [16]

$$\mathbf{h}_{m,u}^{\text{LoS}}(t) = \Big[1,\ e^{-j\frac{2\pi}{\lambda}d_u \sin\left(\theta_{m,u}(t)\right)\cos\left(\phi_{m,u}(t)\right)},\ \ldots,\ e^{-j\frac{2\pi}{\lambda}d_u(N-1)\sin\left(\theta_{m,u}(t)\right)\cos\left(\phi_{m,u}(t)\right)}\Big]^T, \tag{8}$$

where $\lambda = \frac{c}{f_c}$ is the wavelength of the signal, and d_u is the spacing between adjacent RIS elements. $\phi_{m,u}(t)$ is the azimuth angle at the UCR and $\theta_{m,u}(t)$ is the elevation angle at UCR. Moreover, $[\cdot]^T$ indicates the transpose operation.

2) Channel between the UCR and BS: We consider the channel from the UCR to the BS, which is equipped with K antennas. The channel matrix $\mathbf{H}_{u,bs}(t) \in \mathbb{C}^{K\times N}$ can be expressed as [14]

$$\mathbf{H}_{u,bs}(t) = \sqrt{\left(\frac{4\pi f_c d_{u,bs}(t)}{c}\right)^{-\Omega}} \left(\sqrt{\frac{\gamma}{\gamma+1}}\mathbf{H}_{u,bs}^{\text{LoS}}(t)\right), \tag{9}$$

where $d_{u,bs}(t)$ is the distance between the UCR and the BS which can be computed as $d_{u,bs}(t) = ((x_{bs} - x_u(t))^2 + (y_{bs} - y_u(t))^2 + (L - H)^2)^{1/2}$. Moreover, the LoS component of the channel matrix, $\mathbf{H}_{u,bs}^{\text{LoS}}(t) \in \mathbb{C}^{K\times N}$, is expressed as $\mathbf{H}_{u,bs}^{\text{LoS}}(t) = \mathbf{a}_{bs}\left(\theta_{u,bs}(t), \phi_{u,bs}(t)\right)\mathbf{a}_u^H\left(\theta_{u,bs}(t), \phi_{u,bs}(t)\right)$, where $\mathbf{a}_{bs}\left(\theta_{u,bs}(t), \phi_{u,bs}(t)\right) \in \mathbb{C}^{K\times 1}$ denotes the steering vector of the BS, and $\mathbf{a}_u\left(\theta_{u,bs}(t), \phi_{u,bs}(t)\right) \in \mathbb{C}^{N\times 1}$ denotes the steering vector of the UCR. Therefore, $\mathbf{a}_{bs}(\theta_{u,bs}(t), \phi_{u,bs}(t))$ can be expressed as [16]

$$\mathbf{a}_{bs}\left(\theta_{u,bs}(t), \phi_{u,bs}(t)\right) = \Big[1,\ e^{-j\frac{2\pi}{\lambda}d_{bs}\sin\left(\theta_{u,bs}(t)\right)\cos\left(\phi_{u,bs}(t)\right)},\ \ldots,\ e^{-j\frac{2\pi}{\lambda}d_{bs}(K-1)\sin\left(\theta_{u,bs}(t)\right)\cos\left(\phi_{u,bs}(t)\right)}\Big]^T, \tag{10}$$

where d_{bs} is the spacing between the BS antennas. Similarly, $\mathbf{a}_u(\theta_{u,bs}(t), \phi_{u,bs}(t))$ can be calculated as [16]

$$\mathbf{a}_u\left(\theta_{u,bs}(t), \phi_{u,bs}(t)\right) = \Big[1,\ e^{-j\frac{2\pi}{\lambda}d_u\sin\left(\theta_{u,bs}(t)\right)\cos\left(\phi_{u,bs}(t)\right)},\ \ldots,\ e^{-j\frac{2\pi}{\lambda}d_u(N-1)\sin\left(\theta_{u,bs}(t)\right)\cos\left(\phi_{u,bs}(t)\right)}\Big]^T, \tag{11}$$

where $\phi_{u,bs}(t)$ and $\theta_{u,bs}(t)$ represent the azimuth and elevation angles at the BS, respectively. Accordingly, the cascaded channel between the m-th user and the BS can be expressed as $\mathbf{g}_{m,bs}(t) = \mathbf{H}_{u,bs}(t)\mathbf{\Theta}_u(t)\mathbf{h}_{m,u}(t)$. Here, we assume a LoS-dominant UCR link. This reflects typical maritime conditions with an elevated UAV and an open sea surface. It also keeps the model tractable.

3) URLLC user to LEO satellite: The communication link between the m-th user and the LEO satellite is modeled using free-space path loss as the path loss model. Therefore, the channel vector $\mathbf{h}_{m,sat}(t) \in \mathbb{C}^{Y\times 1}$ between the m-th user and the LEO satellite can be formulated as [14]

$$\mathbf{h}_{m,sat}(t) = \sqrt{\left(\frac{4\pi f_c d_{m,sat}(t)}{c}\right)^{-\Omega}} \left(\sqrt{\frac{\gamma}{\gamma+1}}\mathbf{h}_{m,sat}^{\text{LoS}}(t) + \sqrt{\frac{1}{\gamma+1}}\mathbf{h}_{m,sat}^{\text{NLoS}}(t)\right), \tag{12}$$

where $\mathbf{h}_{m,sat}^{\text{LoS}}(t) \in \mathbb{C}^{Y\times 1}$ and $\mathbf{h}_{m,sat}^{\text{NLoS}}(t) \in \mathbb{C}^{Y\times 1}$ denote the LoS and NLoS components, respectively. The distance $d_{m,sat}(t)$ between the m-th user and the LEO satellite can be calculated as $d_{m,sat}(t) = ((x_{sat}(t) - x_m(t))^2 + (y_{sat}(t) - y_m(t))^2 + r^2)^{1/2}$.

D. Communication Model

1) URLLC user to BS communication: URLLC users are enabled to send data to the BS through UCR. At the BS, the signal-to-noise ratio (SNR) of the m-th user can be expressed as [21]

$$\gamma_m^{\text{bs}}(t) = \frac{\mathcal{P}_m(t)||\mathbf{g}_{m,bs}(t)||^2}{\sigma_{bs}^2(t)}, \tag{13}$$

where $\mathcal{P}_m(t)$ represents the transmit power of user m and $\sigma(t)$ 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 [22]

$$\mathcal{R}_m^{\text{bs}}(t) \approx \beta_m(t)\mathcal{B}\left(\log_2\left(1+\gamma_m^{\text{bs}}(t)\right) - \sqrt{\frac{V_m^{\text{bs}}(t)}{N_{bs}}}\frac{Q^{-1}(\epsilon_m^{\text{bs}})}{\ln 2}\right), \tag{14}$$

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

2) URLLC user to satellite communication: When users are outside the coverage of the UCR, they transmit data to an available LEO satellite. Consequently, the SNR of the m-th user can be calculated as [21]

$$\gamma_m^{\text{sat}}(t) = \frac{\mathcal{P}_m(t)||\mathbf{h}_{m,sat}(t)||^2}{\sigma_{sat}^2(t)}, \tag{15}$$

where $\sigma_{sat}(t)$ is the instantaneous noise power, characterized by a Gaussian complex normal distribution $\sim \mathcal{CN}\left(0, \sigma^2\right)$. Therefore, the data rate of the m-th user can be expressed as [22]

$$\mathcal{R}_m^{\text{sat}}(t) \approx \beta_m(t)\mathcal{B}\left(\log_2\left(1+\gamma_m^{\text{sat}}(t)\right) - \sqrt{\frac{V_m^{\text{sat}}(t)}{N_{sat}}}\frac{Q^{-1}(\epsilon_m^{\text{sat}})}{\ln 2}\right), \tag{16}$$

where N_{sat} is the length of the block. The parameter ϵ_m^{sat} corresponds to the probability of decoding errors. Moreover, $V_m^{\text{sat}}(t)$ denotes the channel dispersion, which can be expressed as $V_m^{\text{sat}}(t) = 1 - \left[1+\gamma_m^{\text{sat}}(t)\right]^{-2}$.

E. Data Processing Model

In our model, we consider computation-intensive and latency-sensitive sensor data from the m-th user as a three-tuple $J_m = \{D_m, C_m, \tau_m^{\max}\}$, where D_m denotes the total data size (in bits), C_m represents the overall computational complexity (in CPU cycles), and $\tau_m^{\max}$ is the maximum tolerable latency (in seconds). To facilitate flexible offloading and efficient resource allocation under the URLLC, the task J_m is decomposed into L_m sub tasks, which can be denoted as $J_m = \{J_{m,1}, J_{m,2}, \ldots, J_{m,L_m}\}$, where each subtask is characterized by its own three-tuple $J_{m,\ell} = \{D_{m,\ell}, C_{m,\ell}, \tau_{m,\ell}^{\max}\}$ for $\ell = 1, \ldots, L_m$. The split of sub tasks ensures that the total data size and computational complexity of all sub tasks collectively equal those of the original task, represented as $\sum_{\ell=1}^{L_m} D_{m,\ell} = D_m$ and $\sum_{\ell=1}^{L_m} C_{m,\ell} = C_m$ [23].

1) Data transmission and caching: Let, define $q_m(t) \in \{0,1\}$ as a binary indicator for the transmission path based on distance from the UCR to the m-th user. The parameter $d_{\min}$ represents the minimum threshold distance. If horizontal distance between user m and UCR $((x_u(t) - x_m(t))^2 + (y_u(t) - y_m(t))^2)^{1/2} > d_{\min}$, the user is outside the UCR's coverage, and the m-th user ℓ-th sub task is transmitted to the LEO satellite, represented by $q_m(t) = 0$. Conversely, if $((x_u(t) - x_m(t))^2 + (y_u(t) - y_m(t))^2)^{1/2} \leq d_{\min}$, the user is within the UCR's coverage, and the m-th user ℓ-th sub task is transmitted to the BS, denoted by $q_m(t) = 1$. Moreover, we define two binary variables for the caching decision at the BS and the LEO satellite. Let $z_{m,\ell}^{\text{bs}}(t) \in \{0,1\}$ represent the caching variable at the BS and $z_{m,\ell}^{\text{sat}}(t) \in \{0,1\}$ represent the caching variable at the LEO satellite. If $z_{m,\ell}^{\text{bs}}(t) = 1$, the m-th user ℓ-th sub task is cached at the BS. Otherwise, if $z_{m,\ell}^{\text{bs}}(t) = 0$, the m-th user ℓ-th sub task is not cached at the BS. Similarly, if $z_{m,\ell}^{\text{sat}}(t) = 1$, the m-th user ℓ-th sub task is cached at the LEO satellite, whereas if $z_{m,\ell}^{\text{sat}}(t) = 0$, the m-th user ℓ-th sub task is not cached at the LEO satellite.

2) Latency model: Let $\alpha_{m,\ell}(t) \in [0,1]$ denote the fraction of the m-th user ℓ-th subtask that is transmitted to the BS or LEO satellite. The local processing latency of the m-th user ℓ-th subtask can be calculated as

$$T_{m,\ell}^{\text{local}} = \frac{(1-\alpha_{m,\ell}(t))C_{m,\ell}}{f_m^{\text{local}}(t)}, \tag{17}$$

where, $f_m^{\text{local}}(t)$ is the computational frequency of the m-th user's processor at time t. Furthermore, the transmission latency to BS through UCR for the m-th user l-th subtask can be expressed as

$$T_{m,\ell}^{\text{tr,bs}} = \frac{\alpha_{m,\ell}(t)D_{m,\ell}}{\mathcal{R}_m^{\text{bs}}(t)}. \tag{18}$$

Similarly, the transmission latency of the m-th user l-th subtask to LEO satellite can be formulated as

$$T_{m,\ell}^{\text{tr,sat}} = \frac{\alpha_{m,\ell}(t)D_{m,\ell}}{\mathcal{R}_m^{\text{sat}}(t)}. \tag{19}$$

Therefore, total transmission latency of the m-th user l-th subtask can be expressed as

$$T_{m,\ell}^{\text{tr,tot}} = q_m(t)T_{m,\ell}^{\text{tr,bs}} + (1-q_m(t))T_{m,\ell}^{\text{tr,sat}}. \tag{20}$$

Furthermore, the processing latency of the m-th user l-th subtask at the BS can be formulated as

$$T_{m,\ell}^{\text{pr,bs}} = \frac{\alpha_{m,\ell}(t)C_{m,\ell}}{f^{\text{bs}}(t)}, \tag{21}$$

where $f^{\text{bs}}(t)$ is computational capacity of BS processor. Additionally, the processing latency of the m-th user l-th subtask at the LEO satellite can be formulated as

$$T_{m,\ell}^{\text{pr,sat}} = \frac{\alpha_{m,\ell}(t)C_{m,\ell}}{f^{\text{sat}}(t)}, \tag{22}$$

where, $f^{\text{sat}}(t)$ is the computational frequency of the satellite processor. Therefore, total end-to-end latency of the m-th user l-th subtask can be calculated as

$$\begin{aligned} T_{m,\ell}^{\text{tot}} = {} & T_{m,\ell}^{\text{local}} + q_m(t)(1-z_{m,\ell}^{\text{bs}}(t))T_{m,\ell}^{\text{tr,bs}} \\ & + (1-q_m(t))(1-z_{m,\ell}^{\text{sat}}(t))T_{m,\ell}^{\text{tr,sat}} \\ & + q_m(t)T_{m,\ell}^{\text{pr,bs}} + (1-q_m(t))T_{m,\ell}^{\text{pr,sat}}. \end{aligned} \tag{23}$$

TABLE I
SUMMARY OF KEY NOTATIONS.

Notation	Definition	Notation	Definition
M	Number of URLLC users	N	Number of passive RIS elements
K	Number of antennas at BS	Y	Number of antennas at LEO satellite
L	Altitude of UCR	H	Altitude of BS antennas
$d_{m,u}(t)$	Distance from user m to UCR	$d_{u,bs}(t)$	Distance from UCR to BS
$d_{m,sat}(t)$	Distance from user m to LEO satellite	$\psi_m(t)$	Heading angle of m-th user
$\rho_u(t)$	Heading angle of UCR	Ω	Path loss exponent
γ	Rician factor	f_c	Carrier frequency
$\mathcal{B}$	System bandwidth	$\mathcal{P}_m(t)$	Transmission power of user m
$\beta_m(t)$	System bandwidth allocation factor	D_m	Task size which is generated by user m
C_m	Task complexity which is generated by user m	$\tau_m^{\max}$	Maximum tolerable latency for the task of user m
$D_{m,\ell}$	Subtask size of user m	$C_{m,\ell}$	Subtask complexity which is generated by user m
$\tau_{m,\ell}^{\max}$	Maximum tolerable latency for the subtask of user m	$d_{\min}$	Minimum distance for UCR coverage
$\alpha_{m,\ell}(t)$	Offloading fraction variable	$z_{m,\ell}^{\text{bs}}(t)$	Caching variable at BS
$q_m(t)$	Variable for determining the transmission path	$z_{m,\ell}^{\text{sat}}(t)$	Caching variable at LEO satellite
ξ_{bs}	Energy coefficient of the processor of BS-MEC server	ξ_{sat}	Energy coefficient of the processor of LEO satellite MEC server
ξ_{local}	Energy coefficient of the processor of local user processor	r	Distance from Earth surface to LEO satellite
$T_{m,\ell}^{\text{local}}$	Local processing latency for m-th user ℓ-th subtask	$T_{m,\ell}^{\text{tr,bs}}$	Transmission latency to BS of m-th user ℓ-th subtask
$T_{m,\ell}^{\text{tr,sat}}$	Transmission latency to LEO satellite	$T_{m,\ell}^{\text{tr,tot}}$	Total transmission latency of m-th user ℓ-th subtask
$T_{m,\ell}^{\text{pr,bs}}$	Processing latency at BS for m-th user ℓ-th subtask	$T_{m,\ell}^{\text{pr,sat}}$	Processing latency at LEO satellite for m-th user ℓ-th subtask
$T_{m,\ell}^{\text{tot}}$	Total end-to-end latency of m-th user ℓ-th subtask	$f_m^{\text{local}}(t)$	m-th user device computational power
$f^{\text{bs}}(t)$	BS-MEC computational power for m-th user	$f^{\text{sat}}(t)$	LEO satellite MEC computational power for m-th user
$E_{m,\ell}^{\text{local}}(t)$	Local processing energy consumption	$E_{m,\ell}^{\text{tr,bs}}(t)$	Energy consumption for transmission to BS
$E_{m,\ell}^{\text{tr,sat}}(t)$	Energy consumption for transmission to LEO satellite	$E_{m,\ell}^{\text{pr,bs}}(t)$	Processing energy consumption at BS-MEC
$E_{m,\ell}^{\text{pr,sat}}(t)$	Processing energy consumption at LEO satellite	$E_{m,\ell}^{\text{tot}}(t)$	Total energy consumption of m-th user ℓ-th subtask

3) Energy model: The local processing energy consumption of the m-th user ℓ-th subtask can be calculated as

$$E_{m,\ell}^{\text{local}}(t) = \xi_{\text{local}}(1-\alpha_{m,\ell}(t))C_{m,\ell}(f_m^{\text{local}}(t))^2, \tag{24}$$

where ξ_{local} denotes the energy coefficient associated with the m-th user's processor, determined by the capacitance of the integrated circuit. Besides, the transmission energy consumption to BS through UCR for the m-th user l-th subtask can be expressed as

$$E_{m,\ell}^{\text{tr,bs}}(t) = \mathcal{P}_m(t)T_{m,\ell}^{\text{tr,bs}}. \tag{25}$$

Similarly, the transmission energy consumption of the l-th subtask from the m-th user to the LEO satellite can be calculated as

$$E_{m,\ell}^{\text{tr,sat}}(t) = \mathcal{P}_m(t)T_{m,\ell}^{\text{tr,sat}}. \tag{26}$$

Therefore, end-to-end total transmission energy consumption of the m-th user l-th subtask can be expressed as

$$E_{m,\ell}^{\text{tr,tot}}(t) = q_m(t)E_{m,\ell}^{\text{tr,bs}}(t) + (1-q_m(t))E_{m,\ell}^{\text{tr,sat}}(t). \tag{27}$$

Furthermore, the processing energy consumption of the m-th user l-th subtask at the BS can be formulated as

$$E_{m,\ell}^{\text{pr,bs}}(t) = \xi_{\text{bs}}\alpha_{m,\ell}(t)C_{m,\ell}(f^{\text{bs}}(t))^2, \tag{28}$$

where ξ_{bs} denotes the energy coefficient of the BS-MEC processor, determined by the capacitance of the integrated chip. Similarly, the processing energy consumption of the l-th subtask from the m-th user at the LEO satellite can be calculated as

$$E_{m,\ell}^{\text{pr,sat}}(t) = \xi_{\text{sat}}\alpha_{m,\ell}(t)C_{m,\ell}(f^{\text{sat}}(t))^2, \tag{29}$$

where ξ_{sat} is the energy coefficient of the LEO satellite-MEC processor, which depends on the capacitance of the integrated chip. Therefore, the total energy consumption of the m-th user l-th subtask can be calculated as

$$\begin{aligned} E_{m,\ell}^{\text{tot}}(t) = {} & E_{m,\ell}^{\text{local}}(t) + q_m(t)(1-z_{m,\ell}^{\text{bs}}(t))E_{m,\ell}^{\text{tr,bs}}(t) \\ & + (1-q_m(t))(1-z_{m,\ell}^{\text{sat}}(t))E_{m,\ell}^{\text{tr,sat}}(t) \\ & + q_m(t)E_{m,\ell}^{\text{pr,bs}}(t) + (1-q_m(t))E_{m,\ell}^{\text{pr,sat}}(t). \end{aligned} \tag{30}$$

F. Problem Formulation

In this paper, we aim to jointly optimize the task offloading fractions, transmission power, bandwidth allocation, local computational resource usage, and caching decisions to minimize the total system cost in our proposed maritime 6G-enhanced SAGIN. The system cost reflects a trade-off between energy consumption and latency, which arises from both communication and computation processes distributed across all SAGIN layers. Based on this formulation, the objective function can be defined as

$$\mathcal{F}(\mathcal{X}) = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{\ell=1}^{L_m}\left(w_1E_{m,\ell}^{\text{tot}}(t) + w_2T_{m,\ell}^{\text{tot}}(t)\right) \tag{31}$$

where $\mathcal{X} \triangleq \{\alpha_{m,\ell}(t), \mathcal{P}_m(t), \beta_m(t), f_m^{\text{loc}}(t), z_{m,\ell}^{\text{bs}}(t), z_{m,\ell}^{\text{sat}}(t)\}$. The $\alpha_{m,\ell}(t) \in [0,1]$ denotes the fraction of subtask ℓ offloaded by user m at time t. $\mathcal{P}_m(t)$ is the transmit power allocation of user m. $\beta_m(t) \in [0,1]$ is the fraction of total bandwidth allocated to user m at time t. $f_m^{\text{loc}}(t)$ is the allocated CPU processing frequency at the local device. Moreover, $z_{m,\ell}^{\text{bs}}(t), z_{m,\ell}^{\text{sat}}(t) \in \{0,1\}$ indicate whether subtask ℓ of user m is cached at the BS or the LEO satellite. Therefore, the optimization problem can be formulated as follows:

$$
\begin{aligned}
(\mathbf{P1}): \quad & \min_{\mathcal{X}} \ \mathcal{F}(\mathcal{X}) && && \text{(32a)}\\
\text{s.t.} \quad & T_{m,\ell}^{\text{tot}}(t) \le \tau_{m,\ell}^{\max}, && \forall m,\ell,t, && \text{(32b)}\\
& \sum_{\ell=1}^{L_m} D_{m,\ell} = D_m, \quad \sum_{\ell=1}^{L_m} C_{m,\ell} = C_m, && \forall m, && \text{(32c)}\\
& 0 \le \alpha_{m,\ell}(t) \le 1, && \forall m,\ell,t, && \text{(32d)}\\
& 0 < \mathcal{P}_m(t) \le \mathcal{P}_m^{\max}, && \forall m,t, && \text{(32e)}\\
& \sum_{m=1}^{M} \beta_m(t) \le 1, \quad \beta_m(t) > 0, && \forall t, && \text{(32f)}\\
& 0 < f_m^{\text{local}}(t) \le f_{m,\text{loc}}^{\max}, && \forall m,t, && \text{(32g)}\\
& \sum_{m:q_m(t)=1} \sum_{\ell=1}^{L_m} z_{m,\ell}^{\text{bs}}(t) D_{m,\ell} \le S_{\text{bs}}^{\max}, && \forall t, && \text{(32h)}\\
& \sum_{m:q_m(t)=0} \sum_{\ell=1}^{L_m} z_{m,\ell}^{\text{sat}}(t) D_{m,\ell} \le S_{\text{sat}}^{\max}, && \forall t, && \text{(32i)}\\
& z_{m,\ell}^{\text{bs}}(t), z_{m,\ell}^{\text{sat}}(t) \in \{0,1\}, && \forall m,\ell,t, && \text{(32j)}\\
& \mathcal{R}_m^{\text{bs}}(t) \ge \mathcal{R}_m^{\min}, \quad \mathcal{R}_m^{\text{sat}}(t) \ge \mathcal{R}_m^{\min}, && \forall m,t, && \text{(32k)}\\
& z_{m,\ell}^{\text{bs}}(t) + z_{m,\ell}^{\text{sat}}(t) \le 1, && \forall m,\ell,t. && \text{(32l)}
\end{aligned}
$$

As specified in (32), the optimization problem aims to minimize the total cost for all users and their respective subtasks during the operational time of the harbor by optimizing the transmission power, bandwidth allocation, computational resource allocation, and caching decisions subject to constraints. The constraint (32b) ensures that the total latency for each subtask of user m does not exceed the maximum tolerable latency threshold $\tau_{m,\ell}^{\max}$ which maintains stringent latency requirements. Constraint (32c) enforces that the total data size and computational complexity of all subtasks collectively sum to the original task size and complexity. Constraint (32d) ensures that the offloading fraction $\alpha_{m,\ell}(t)$ remains within the range $[0,1]$. Constraint (32e) enforces that the transmission power allocated to each user does not exceed the maximum power limit $\mathcal{P}_m^{\max}$ and ensure energy-efficient communication. The bandwidth allocation is governed by constraint (32f), where each user's allocated bandwidth fraction $\beta_m(t)$ must be non-negative, and the total allocated bandwidth across all users does not exceed the available system bandwidth. Constraint (32g) ensures that the local computation frequency for each user does not exceed its maximum processing capability $f_{m,\text{loc}}^{\max}$ and prevents computational overload. Constraints (32h) and (32i) impose caching constraints on the BS and LEO satellite. It ensure the total amount of cached data does not exceed the storage capacity limits $S_{\text{bs}}^{\max}$ and $S_{\text{sat}}^{\max}$, respectively. Constraint (32j) enforces the binary nature of the caching decision variables $z_{m,\ell}^{\text{bs}}(t)$ and $z_{m,\ell}^{\text{sat}}(t)$ and ensure that a subtask is either cached or not. Constraint (32k) ensures that the achievable data rate for each user through both the BS and the LEO satellite remains above a predefined minimum threshold $\mathcal{R}_m^{\min}$. Constraint (32l) ensures that a subtask is cached at most at one location, either the BS or the LEO satellite.

III. Proposed Solution

The optimization problem formulated in (32) is a MINLP problem involving continuous and binary decision variables. These include the bandwidth allocation $\beta_m(t)$, the offloading fraction $\alpha_{m,l}(t)$, the transmission power allocation $\mathcal{P}_m(t)$, the local processing frequency $f_m^{\text{loc}}(t)$, and the binary caching decisions $z_{m,\ell}^{\text{bs}}(t)$ and $z_{m,\ell}^{\text{sat}}(t)$. The coexistence of discrete and continuous actions, combined with nonlinear coupling among system resources, creates a large and complex search space that is difficult to address using conventional optimization techniques. In order to address these challenges, we propose a quantum-enhanced DRL framework that leverages quantum feature encoding to efficiently represent high-dimensional state-action spaces. Specifically, we employ quantum-enhanced deep deterministic policy gradient (QEDDPG) and quantum-enhanced proximal policy optimization (QEPPO) to jointly manage continuous and discrete decision variables. These approaches aim to dynamically minimize the total system cost, defined as the combination of energy consumption and task latency, providing a practical and scalable solution to resource allocation in the maritime 6G-enhanced SAGIN.

A. Transformation to Quantum-Enhanced Deep Reinforcement Learning Framework

To apply the proposed quantum-enhanced DRL frameworks for solving the optimization problem defined in (32), we model the environment as a Markov decision process (MDP) characterized by the tuple $\mathcal{S}, \mathcal{A}, \mathcal{R}$. Here, $\mathcal{S}$ represents the state space, which captures the observable environment features upon which decisions are based. The action space $\mathcal{A}$ outlines the complete set of actions available to the agent for influencing system dynamics. Meanwhile, the reward function $\mathcal{R}$ quantifies the feedback resulting from each action, providing the necessary learning signal that drives policy improvement. At every time step t, the agent interacts with the environment by first observing the current state $s(t)$, then choosing an action $a(t)$, and finally receiving a reward $r(t)$ that reflects the immediate outcome of its decision. This feedback loop allows the agent to iteratively refine its policy through continuous interaction with the maritime 6G-enhanced SAGIN system.

`Observation Space`: The state space $\mathcal{S}$ represents the system status at each time slot t. It provides the quantum-enhanced agent with the necessary information for decision-making. Hence, the state $s(t)$ at time t is defined as $s(t) = \left[E_{\text{norm}}(t), T_{\text{norm}}(t), \overline{\gamma}_{\text{bs}}(t), \overline{\gamma}_{\text{sat}}(t), r_{\text{rem}}(t), T_{\text{tr}}^{\text{norm}}(t)\right]$,

where $E_{\text{norm}}(t)$ is the normalized cumulative energy consumption at time t, and $T_{\text{norm}}(t)$ is the normalized cumulative latency. Moreover, $\overline{\gamma}_{\text{bs}}(t)$ represents the average SINR from the users to the base station, while $\overline{\gamma}_{\text{sat}}(t)$ represents the average SNR from the users to the LEO satellite. Furthermore, $r_{\text{rem}}(t)$ is the fraction of remaining time slots relative to the total operation period, providing temporal urgency information and $T_{\text{tr}}^{\text{norm}}(t)$ is the normalized average transmission latency, which quantifies the average transmission latency experienced by the users at time t.

`Action Space`: The action space $\mathcal{A}$ defines the agent's control decisions at each time slot t. It determines the adjustable system variables that influence task offloading, power allocation, bandwidth allocation, local computation, and caching strategies. Therefore, the action space $a(t)$ at time t can be defined as $a(t) = \left[\alpha_{m,\ell}(t), \mathcal{P}_m(t), \beta_m(t), f_m^{\text{local}}(t), z_{m,\ell}^{\text{bs}}(t), z_{m,\ell}^{\text{sat}}(t)\right]$, where $\alpha_{m,\ell}(t) \in [0,1]$ denotes the offloading fraction of subtask ℓ generated by user m at time t. $\mathcal{P}_m(t)$ represents the transmission power allocated to user m. Moreover, $\beta_m(t) \in [0,1]$ denotes the fraction of system bandwidth assigned to user m, and $f_m^{\text{local}}(t)$ denotes the local CPU processing frequency allocated to user m at time t. Furthermore, $z_{m,\ell}^{\text{bs}}(t) \in \{0,1\}$ and $z_{m,\ell}^{\text{sat}}(t) \in \{0,1\}$ are binary caching decision variables indicating whether the ℓ-th subtask of user m is cached at the base station or at the LEO satellite, respectively.

`Reward Function`: The reward function $\mathcal{R}$ plays a pivotal role in reinforcing the learning behavior of the agent by continuously evaluating its actions. It provides real-time feedback that guides the policy toward desirable outcomes by measuring how each decision influences system performance. Specifically, it captures the immediate consequences of actions taken in dynamic environments, enabling the agent to improve its decision-making over time. Accordingly, the reward per step at time t, denoted by $r(t)$, can be formulated as

$$r(t) = -\frac{1}{W}\left(\sum_{m=1}^{M}\sum_{\ell=1}^{L_m}\left(w_1 E_{m,\ell}^{\text{tot}}(t) + w_2 T_{m,\ell}^{\text{tot}}(t)\right) + P(t)\right), \quad (33)$$

where $\mathcal{F}(\mathcal{X})$ denotes the system cost, and $P(t)$ represents the penalty terms introduced to enforce the constraints specified in (32b) to (32l). Moreover, W represents the scaling factor of the reward function. A negative reward structure is adopted to encourage the agent to minimize the overall system cost, thereby promoting energy-efficient and low-latency operation within the maritime 6G-enabled SAGIN.

B. Introduction to Quantum Computing

QC marks a significant technological breakthrough in contemporary science and engineering. Unlike classical computing, which uses binary bits, it works with quantum bits (qubits). Qubits use the principle of superposition, allowing them to exist in multiple states at once. This capability, combined with quantum entanglement and inherent parallelism, enables quantum systems to address specific classes of problems more efficiently than traditional computing architectures. Moreover, a qubit can exist in a combination of two basis states. Therefore, we can express a single qubit as a linear combination of computational basis states. Thus, it can be represented as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$ and satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. This condition ensures that the qubit represents a valid quantum superposition of both logical states. Upon measurement, the superposition collapses to either $|0\rangle$ or $|1\rangle$, with the associated probabilities which can be expresed as

$$P(|0\rangle) = |\alpha|^2, \qquad P(|1\rangle) = |\beta|^2.$$

Furthermore, quantum computers manipulate qubit states using quantum gates. These gates operate differently from classical logic gates. Moreover, these gates are described by unitary matrices, ensuring that operations are reversible and norm-preserving. Quantum gates act on individual or multiple qubits and serve as the building blocks of quantum circuits. Their composition enables the construction of algorithms capable of solving classically intractable problems. The basic quantum gates are summarized in the Table II.

TABLE II
BASIC QUANTUM GATES [25]

Gate	Matrix	Functionality
Hadamard (**H**)	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$	Generates superposition of $\|0\rangle$ and $\|1\rangle$
Pauli-X (**X**)	$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$	Bit-flip: $\|0\rangle \leftrightarrow \|1\rangle$
Pauli-Y (**Y**)	$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$	Applies bit and phase flip (Y-axis rotation)
Pauli-Z (**Z**)	$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$	Applies phase flip (Z-axis rotation)
RX(θ)	$\begin{bmatrix}\cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2})\\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2})\end{bmatrix}$	Rotates qubit around X-axis by angle θ
RY(θ)	$\begin{bmatrix}\cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2})\\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2})\end{bmatrix}$	Rotates qubit around Y-axis by angle θ
RZ(θ)	$\begin{bmatrix}e^{-i\theta/2} & 0\\ 0 & e^{i\theta/2}\end{bmatrix}$	Rotates qubit around Z-axis by angle θ
CNOT	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$	Flips target qubit if control qubit is $\|1\rangle$

C. Preliminaries of Quantum Machine Learning

QML is a growing research direction that integrates the strengths of quantum computing with classical machine learning techniques. As illustrated in Fig 2, the process begins with data encoding, where classical input values are transformed into quantum states to be processed by quantum circuits. This transformation is carried out using parameterized quantum gates, such as rotation gates (**RX**, **RY**, **RZ**), and entanglement operations that link qubits to capture complex patterns in the data [44]. There are various data encoding strategies such as amplitude encoding, angle encoding, and basis encoding, which significantly influence the model's expressiveness and circuit depth.

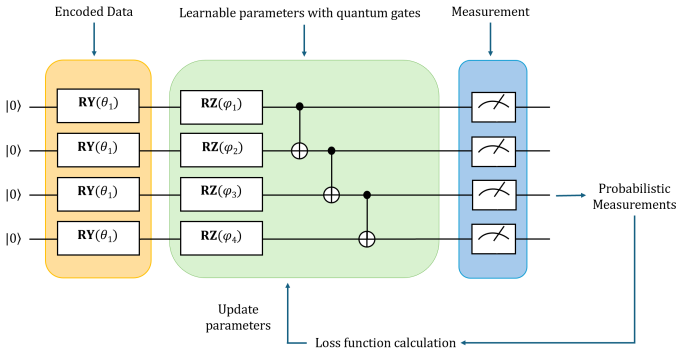

Fig. 2. QML pipeline with data encoding, parametric quantum circuit and quantum measurement.

1) Amplitude encoding: Amplitude encoding maps a normalized classical vector $\mathbf{x} = [x_0, x_1, \ldots, x_{2^n-1}]$ into the amplitude components of a quantum state distributed across n qubits. This enables a concise quantum representation of classical data [45]

$$|\psi\rangle = \sum_{i=0}^{2^n-1} x_i|i\rangle, \quad \text{with} \quad \sum |x_i|^2 = 1.$$

This encoding scheme enables exponential compression of classical information and make it highly efficient for high-dimensional data representation. However, state preparation often requires deep circuits and significant resources, especially when working with arbitrary vectors.

2) Angle encoding: Angle encoding, also referred to as rotation-based or parametric encoding. This encodes classical features into the angular parameters of single-qubit quantum gates. Specifically, each classical scalar $x \in \mathbb{R}$ is embedded into a qubit using rotational operations, typically around the Y or Z axis. A common implementation uses the **RY** gate as follows [45]

$$|\psi\rangle = \mathbf{RY}(2x)|0\rangle = \cos(x)|0\rangle + \sin(x)|1\rangle$$

This approach offers a hardware-efficient encoding strategy and it is well-suited for near-term quantum devices.

3) Basic encoding: Basis encoding is the most direct method for representing classical binary information in quantum systems. It maps an n-bit classical string $b = b_0 b_1 \ldots b_{n-1}$, where $b_i \in \{0, 1\}$, into a quantum state using the computational basis which can be expressed as

$$|\psi\rangle = |b_0 b_1 \ldots b_{n-1}\rangle.$$

This method is efficient for discrete data. However, it lacks scalability for continuous values and offers limited quantum advantage.

After encoding process, the quantum states are processed through a QNN. A QNN is composed of layers of quantum gates arranged to simulate the behavior of neurons and activation functions. These networks are designed using parameterized unitary operations and often trained through variational algorithms. The QNN serves as the core computational block where learning takes place by adjusting gate parameters to minimize a loss function which is similar to classical deep learning [45].

D. Quantum-Enhanced Deep Deterministic Policy Gradient Algorithm

We employ the QEDDPG framework, a quantum-enhanced, model-free DRL method. Here the classical states $s(t)$ are first encoded into quantum states $|s_q(t)\rangle$ through higher order encoding (HOE). This quantum feature map captures both linear and nonlinear correlations within the state representation and enable richer expressiveness and improved policy learning. The HOE method constructs an n-qubit circuit to embed the normalized classical input. Each state element x is first processed through a hyperbolic tangent function and ensure the values lie within $[-1, 1]$. The resulting normalized vector is then transformed into a quantum state through the unitary operator $\mathbf{U}_{\text{HOE}}(x)$, which can be expressed as $|s_q(t)\rangle = \mathbf{U}_{\text{HOE}}(x)|0\rangle^{\otimes n}$. Where, $\mathbf{U}_{\text{HOE}}(x)$ can be denoted as [45]

$$\begin{aligned}\mathbf{U}_{\text{HOE}}(x) = \Bigg(\prod_{i=0}^{n-2} \big[\mathbf{CNOT}_{i,i+1} \cdot \mathbf{RZ}_{i+1}\,(2(\pi - \tanh(x_i))\\ (\pi - \tanh(x_{i+1}))) \cdot \mathbf{CNOT}_{i,i+1}\big]\Bigg) \cdot\\ \left(\bigotimes_{i=0}^{n-1} \mathbf{RZ}_i\,(2\pi \tanh(x_i)) \cdot \mathbf{H}_i\right),\end{aligned} \tag{34}$$

where $\mathbf{H}_i$ is the Hadamard gate on qubit i, $\mathbf{RZ}(\cdot)$ is a Z-axis rotation, and $\mathbf{CNOT}_{i,i+1}$ denotes the controlled-NOT gate between qubits i and $i+1$. Following HOE, two QNNs, also known as parameterized quantum circuits (PQCs), process $|s_q(t)\rangle$. One PQC for the actor network to learn a deterministic policy and another for the critic network to estimate the Q-value. Each PQC comprises multiple variational layers which can be defined by the unitary $\mathbf{U}_{\text{PQC}}(\theta, \phi) = \prod_l \mathbf{U}^{(l)}_{\text{layer}}(\theta^{(l)}, \phi^{(l)})$. Then, $\mathbf{U}^{(l)}_{\text{layer}}(\theta^{(l)}, \phi^{(l)})$ can be expressed as

$$\mathbf{U}^{(l)}_{\text{layer}}(\theta^{(l)}, \phi^{(l)}) = \left(\bigotimes_{i=0}^{n-1} \left(\mathbf{RY}_i(\theta_i^{(l)})\mathbf{RZ}_i(\phi_i^{(l)})\right)\right) \cdot \left(\prod_{i=0}^{n-2} \mathbf{CZ}_{i,i+1}\right), \tag{35}$$

where $\mathbf{RY}(\theta_i^{(l)})$, $\mathbf{RZ}(\phi_i^{(l)})$ are the single-qubit rotations and $\mathbf{CZ}_{i,i+1}$ are controlled-$\mathbf{Z}$ gates for creating the entanglement between adjacent qubits. For the actor PQC, the unitary operator is $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$, with trainable parameters $\theta_a = \{\theta^{(l)}_{a,i}\}$ and $\phi_a = \{\phi^{(l)}_{a,i}\}$. For the critic PQC, the unitary operator is $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)$, with trainable parameters $\theta_c = \{\theta^{(l)}_{c,i}\}$ and $\phi_c = \{\phi^{(l)}_{c,i}\}$. When the actor PQC $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$ is applied to the encoded quantum state $|s_q(t)\rangle$, we measure the policy using projective measurements in the Pauli-$\mathbf{Z}$ basis, obtaining expectation values $\langle \pi_{\theta_a, \phi_a} \rangle = \{\langle \mathbf{Z}_i \rangle\}_{i=0}^{n-1}$. These are averaged over N_{shot} measurements (M) and decoded through $F^{\text{actor}}_{\text{decode}}$ to

compute the deterministic policy which can be expressed as [25]

$$\pi_{\theta_a,\phi_a}(a(t) \mid s(t)) \overset{F_{\text{decode}}^{\text{actor}}}{\longleftarrow} \frac{1}{N_{\text{shot}}} \sum_{l=1}^{N_{\text{shot}}} M(\langle \pi_{\theta_a,\phi_a} \rangle). \tag{36}$$

The decoding operation employs a linear layer with sigmoid activation to produce continuous actions and Gumbel-Softmax logits for binary decisions which forms the deterministic policy $\pi_{\theta_a,\phi_a}(s(t))$. Similarly, the critic PQC $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)$ receives a preprocessed input that combines the state, continuous actions, and discrete action probabilities, and outputs a Q-value $Q_{\theta_c,\phi_c}(s(t), a(t))$ through a linear layer. After applying $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)$, the Q-value is measured using projective measurements in the Pauli-$\mathbf{Z}$ basis, resulting in expectation values $\langle q_{\theta_c,\phi_c} \rangle = \{\langle Z_i \rangle\}_{i=0}^{n-1}$. The expectation values are averaged over N_{shot} measurements and are decoded through $F_{\text{decode}}^{\text{critic}}$ to compute the Q-value which can be calculated as

$$Q_{\theta_c,\phi_c}(s(t), a(t)) \overset{F_{\text{decode}}^{\text{critic}}}{\longleftarrow} \frac{1}{N_{\text{shot}}} \sum_{l=1}^{N_{\text{shot}}} M(\langle q_{\theta_c,\phi_c} \rangle). \tag{37}$$

The decoding operation employs a linear layer that maps the expectation values to a scalar Q-value. Upon executing an action $a(t)$ generated by the deterministic policy $\pi_{\theta_a,\phi_a}(s(t))$ with additive exploration noise in the maritime 6G-enhanced SAGIN environment, the agent receives an immediate reward $r(t)$ and transitions to the next state $s(t+1)$. Then the temporal-difference target $y(t)$ can be expressed as [43], [46]

$$y(t) = r(t) + \gamma Q_{\theta_c',\phi_c'}\left(s(t+1), \pi_{\theta_a',\phi_a'}(s(t+1))\right), \tag{38}$$

where γ denotes the discount factor, and $Q_{\theta_c',\phi_c'}\left(s(t+1), \pi_{\theta_a',\phi_a'}(s(t+1))\right)$ represents the Q-value estimated by the critic PQC using target networks with parameters θ_a', ϕ_a', θ_c', and ϕ_c'. Then, the actor loss $\mathcal{L}_{\text{actor}}$, which aims to maximize the expected Q-value under the current policy, can be formulated as [43], [46]

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}\left[Q_{\theta_c,\phi_c}\left(s(t), \pi_{\theta_a,\phi_a}(s(t))\right)\right]. \tag{39}$$

Therefore, the corresponding policy gradient can be computed as [43], [46]

$$\nabla_{\theta_a,\phi_a}\mathcal{L}_{\text{actor}} = -\mathbb{E}_{s\sim\mathcal{D}}\Bigg[\nabla_{\theta_a,\phi_a}\pi_{\theta_a,\phi_a}(s(t)) \nabla_a Q_{\theta_c,\phi_c}(s(t), a)\Big|_{a=\pi_{\theta_a,\phi_a}(s(t))}\Bigg]. \tag{40}$$

Thus, the critic loss $\mathcal{L}_{\text{critic}}$, which minimizes the TD error between the predicted Q-value and the TD target, can be expressed as [43], [46]

$$\mathcal{L}_{\text{critic}} = \mathbb{E}\left[\left(Q_{\theta_c,\phi_c}(s(t), a(t)) - y(t)\right)^2\right], \tag{41}$$

and its gradient can be computed as [43]

$$\nabla_{\theta_c,\phi_c}\mathcal{L}_{\text{critic}} = \mathbb{E}_{s\sim\mathcal{D}}\Bigg[2\left(Q_{\theta_c,\phi_c}(s(t), a(t)) - y(t)\right) \nabla_{\theta_c,\phi_c} Q_{\theta_c,\phi_c}(s(t), a(t))\Bigg]. \tag{42}$$

The actor and critic parameters are updated using these gradients which are computed through the parameter-shift rule [25], [26]. Moreover, soft target network updates based on Polyak averaging are applied to further stabilize the learning process. The detailed QEDDPG framework is summarized in Algorithm 1.

Algorithm 1 : Proposed Quantum-Enhanced DDPG (QEDDPG) Algorithm for solving (32).

1: Initialize the maritime 6G-enhanced SAGIN environment with its specified parameters.
2: Initialize the actor PQC $\pi(s|\theta_a, \phi_a)$ with parameters θ_a, ϕ_a and the critic PQC $Q(s, a|\theta_c, \phi_c)$ with parameters θ_c, ϕ_c.
3: Initialize the target networks π' and Q' with $\theta_a' \leftarrow \theta_a$, $\phi_a' \leftarrow \phi_a$, $\theta_c' \leftarrow \theta_c$, and $\phi_c' \leftarrow \phi_c$.
4: Set up a replay buffer $\mathcal{D}$.
5: **for** each episode **do**
6: **for** $t = 1, 2, \ldots, T$ **do**
7: Encode the state $s(t)$ into a quantum state $|s_q(t)\rangle$ using HOE as defined in (34).
8: Apply the actor PQC $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$ to $|s_q(t)\rangle$, measure in the Pauli-$\mathbf{Z}$ basis, and decode via $F_{\text{decode}}^{\text{actor}}$ to generate continuous actions $a_{\text{cont}}(t)$ and discrete actions $a_{\text{disc}}(t)$ as $a(t) = \pi(s(t)|\theta_a, \phi_a)$.
9: Generate an action with added noise: $a(t) = \pi(s(t)|\theta_a, \phi_a) + Z\hat{(t)}$.
10: Execute the action $a(t)$, receive reward $r(t)$, and transition to the next state $s(t+1)$.
11: Store $\{s(t), a(t), r(t), s(t+1)\}$ in the replay buffer $\mathcal{D}$.
12: Randomly sample a batch S of transitions from $\mathcal{D}$.
13: Compute target values $y(t)$ for the batch using (38).
14: Update the critic PQC parameters θ_c, ϕ_c by minimizing the loss $\mathcal{L}_{\text{critic}}$ using (41) and its gradient (42).
15: Update the actor PQC parameters θ_a, ϕ_a using the policy gradient from (40).
16: Softly update the target networks using the soft update rule with coefficient τ.
17: **end for**
18: **end for**

E. Quantum-Enhanced Proximal Policy Optimization Algorithm

In this section we propose QEPPO framework, a quantum-powered, model-free DRL strategy. The process begins by encoding the classical state information into quantum states using HOE as defined in (34). As discussed this encoding scheme enable a richer feature representation. Once encoded, the quantum state $|s_q(t)\rangle$ is processed by the actor PQC, denoted as $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$, with trainable parameters $\theta_a = \{\theta_{a,i}^{(l)}\}$ and $\phi_a = \{\phi_{a,i}^{(l)}\}$. The actor PQC applies a sequence of variational layers, where the overall unitary transformation can be defined as

$\mathbf{U}_{\text{PQC}}(\theta,\phi) = \prod_l \mathbf{U}^{(l)}_{\text{layer}}(\theta^{(l)},\phi^{(l)})$, with each individual layer $\mathbf{U}^{(l)}_{\text{layer}}(\theta^{(l)},\phi^{(l)})$ detailed in (35). After processing, projective measurements in the Pauli-$\mathbf{Z}$ basis are performed, yielding a set of expectation values $\langle \pi_{\theta_a,\phi_a}\rangle = \{\langle \mathbf{Z}_i\rangle\}_{i=0}^{n-1}$ that represent the quantum-encoded policy information. These measurement outcomes are averaged across N_{shot} repeated measurements, denoted by M, and decoded through a function $\mathbf{F}^{\text{actor}}_{\text{decode}}$. This produces the final stochastic policy as

$$\pi_{\theta_a,\phi_a}(a(t) \mid s(t)) \overset{\mathbf{F}^{\text{actor}}_{\text{decode}}}{\longleftarrow} \frac{1}{N_{\text{shot}}} \sum_{l=1}^{N_{\text{shot}}} M(\langle \pi_{\theta_a,\phi_a}\rangle). \tag{43}$$

Concurrently, the value estimation is performed using a separate PQC, denoted as $\mathbf{U}_{\text{value}}(\theta_v,\phi_v)$. This also processes $|s_q(t)\rangle$ and is parameterized by $\theta_v = \{\theta^{(l)}_{v,i}\}$ and $\phi_v = \{\phi^{(l)}_{v,i}\}$. After processing, projective measurements in the Pauli-$\mathbf{Z}$ basis are performed. Then the expectation values $\langle v_{\theta_v,\phi_v}\rangle = \{\langle \mathbf{Z}_i\rangle\}_{i=0}^{n-1}$ can be obtained as

$$V_{\theta_v,\phi_v}(s(t)) \overset{\mathbf{F}^{\text{value}}_{\text{decode}}}{\longleftarrow} \frac{1}{N_{\text{shot}}} \sum_{l=1}^{N_{\text{shot}}} M(\langle v_{\theta_v,\phi_v}\rangle). \tag{44}$$

Following the policy sampling, the agent executes the selected action $a(t)$ in the maritime 6G-enhanced SAGIN environment. Then the agent receives an immediate reward $r(t)$, and transitions to the next state $s(t+1)$. Thereafter, the temporal-difference target $y(t)$ is calculated which can be expressed as [47]

$$y(t) = r(t) + \gamma V_{\theta'_v,\phi'_v}(s(t+1)), \tag{45}$$

where γ denotes the discount factor, and $V_{\theta'_v,\phi'_v}(s(t+1))$ represents the state-value estimated by the target value PQC with parameters θ'_v, ϕ'_v. The advantage function, which quantifies the relative benefit of the action $a(t)$, can be computed as [47]

$$A(t) = r(t) + \gamma V_{\theta'_v,\phi'_v}(s(t+1)) - V_{\theta_v,\phi_v}(s(t)), \tag{46}$$

where $V_{\theta_v,\phi_v}(s(t))$ is the value PQC's estimate for the current state with parameters θ_v, ϕ_v. Then, the actor loss $\mathcal{L}_{\text{actor}}$, which aims to optimize the policy, can be formulated as [47]

$$\mathcal{L}_{\text{actor}} = \mathbb{E}\Bigg[\min\Bigg(\frac{\pi_{\theta_a,\phi_a}(a(t) \mid s(t))}{\pi_{\theta_a^{\text{old}},\phi_a^{\text{old}}}(a(t) \mid s(t))} A(t), \text{clip}\Bigg(\frac{\pi_{\theta_a,\phi_a}(a(t) \mid s(t))}{\pi_{\theta_a^{\text{old}},\phi_a^{\text{old}}}(a(t) \mid s(t))}, 1-\epsilon, 1+\epsilon \Bigg) A(t) \Bigg) \Bigg], \tag{47}$$

with its gradient can be computed as [47]

$$\nabla_{\theta_a,\phi_a}\mathcal{L}_{\text{actor}} = \mathbb{E}\Bigg[\nabla_{\theta_a,\phi_a} \min\Bigg(\frac{\pi_{\theta_a,\phi_a}(a(t) \mid s(t))}{\pi_{\theta_a^{\text{old}},\phi_a^{\text{old}}}(a(t) \mid s(t))} A(t), \text{clip}\Bigg(\frac{\pi_{\theta_a,\phi_a}(a(t) \mid s(t))}{\pi_{\theta_a^{\text{old}},\phi_a^{\text{old}}}(a(t) \mid s(t))}, 1-\epsilon, 1+\epsilon \Bigg) A(t) \Bigg) \Bigg], \tag{48}$$

where π_{θ_a,ϕ_a} denotes the combined continuous and discrete policy parameterized by θ_a and ϕ_a, $\pi_{\theta_a^{\text{old}},\phi_a^{\text{old}}}$ represents the policy from the previous iteration, and ϵ denotes the clipping parameter. Moreover, the value loss $\mathcal{L}_{\text{value}}$, which minimizes the error between the predicted and target values, can be expressed as [47]

$$\mathcal{L}_{\text{value}} = \mathbb{E}\left[\left(V_{\theta_v,\phi_v}(s(t)) - y(t) \right)^2 \right], \tag{49}$$

with its gradient can be computed as [47]

$$\nabla_{\theta_v,\phi_v}\mathcal{L}_{\text{value}} = \mathbb{E}\left[2\left(V_{\theta_v,\phi_v}(s(t)) - y(t) \right) \cdot \nabla_{\theta_v,\phi_v} V_{\theta_v,\phi_v}(s(t)) \right]. \tag{50}$$

Similarly, policy and value parameters are updated using parameter-shift rule. Thus, the QEPPO framework is summarized in Algorithm 2.

Algorithm 2 : Proposed Quantum-Enhanced PPO (QEPPO) Algorithm for solving (32).

1: Initialize the maritime 6G-enhanced SAGIN environment with its specified parameters.
2: Initialize the actor PQC $\pi(a|s|\theta_a,\phi_a)$ with parameters θ_a, ϕ_a, and the value PQC $V(s|\theta_v,\phi_v)$ with parameters θ_v, ϕ_v.
3: Set up a rollout buffer $\mathcal{T}$.
4: **for** each iteration **do**
5: **for** $t = 1, 2, \ldots, T$ **do**
6: Encode the state $s(t)$ into a quantum state $|s_q(t)\rangle$ using HOE as defined in (34).
7: Apply the actor PQC $\mathbf{U}_{\text{actor}}(\theta_a,\phi_a)$ to $|s_q(t)\rangle$, measure in the Pauli-$\mathbf{Z}$ basis, and decode via $F^{\text{actor}}_{\text{decode}}$ to generate continuous and discrete actions as $a(t) \sim \pi(a|s(t)|\theta_a,\phi_a)$.
8: Execute the action $a(t)$, receive reward $r(t)$, and transition to the next state $s(t+1)$.
9: Apply the value PQC $\mathbf{U}_{\text{value}}(\theta_v,\phi_v)$ to $|s_q(t)\rangle$, measure in the Pauli-$\mathbf{Z}$ basis, and decode to compute the value $V_{\theta_v,\phi_v}(s(t))$.
10: Store $\{s(t), a(t), r(t), s(t+1), V_{\theta_v,\phi_v}(s(t))\}$ in the $\mathcal{T}$.
11: **end for**
12: Compute target values $y(t)$ and advantages $A(t)$ for the trajectory in $\mathcal{T}$ using (45) and (46).
13: **for** each epoch **do**
14: Compute the actor loss $\mathcal{L}_{\text{actor}}$ over the $\mathcal{T}$ using (47).
15: Update the policy parameters θ_a, ϕ_a by minimizing $\mathcal{L}_{\text{actor}}$ using the gradient $\nabla_{\theta_a,\phi_a}\mathcal{L}_{\text{actor}}$ from (48).
16: Compute the value loss $\mathcal{L}_{\text{value}}$ over the $\mathcal{T}$ using (49).
17: Update the value parameters θ_v, ϕ_v by minimizing $\mathcal{L}_{\text{value}}$ using the gradient $\nabla_{\theta_v,\phi_v}\mathcal{L}_{\text{value}}$ from (50).
18: **end for**
19: Softly update the target value network using the soft update rule with coefficient τ.
20: **end for**

F. Complexity Analysis

The complexity of the proposed QEDDPG algorithm comes from quantum measurements, gradients, replay-buffer sampling, and soft-target updates for its actor and critic PQCs.

Therefore, each step involves HOE with complexity $O(Bn)$, PQC evolution with $O(BL(3n-1))$, and measurement with $O(BnN_{\text{shot}})$. Moreover, classical decoding for the actor covers both continuous and discrete outputs, costing $O(n(d_c+d_d))$, while the critic's linear layer maps quantum outputs to a scalar Q-value with $O(n)$, totaling $O(Bn(d_c+d_d+1))$. Replay sampling from buffer $\mathcal{D}$ costs $O(B)$, and soft updates require $O(N_a+N_c)$, where N_a and N_c denote the actor and critic parameters. Gradients, obtained through the parameter-shift rule, cost $O(BL(3n-1))$, along with $O(Bn(d_c+d_d+1))$ for decoding. Thus, the per-step complexity can be expressed as $O(Bn(L(3n-1)+N_{\text{shot}}+d_c+d_d+1)+N_a+N_c)$. Over T steps and E episodes, the total complexity can be calculated as $O(ET[Bn(L(3n-1)+N_{\text{shot}}+d_c+d_d+1)+N_a+N_c])$.

Similarly, the computational complexity of the proposed QEPPO algorithm mainly arises from the qubit measurement process and the calculation of gradients for its actor and value PQCs. Each step involves HOE calculated as $O(Bn)$, PQC evolution with L layers computed as $O(BL(3n-1))$, and measurement across n qubits with N_{shot} shots, resulting in $O(BnN_{\text{shot}})$. Moreover, classical decoding employs linear mappings for actor continuous ($O(nd_c)$), actor discrete ($O(nd_d)$), and value ($O(n)$) outputs, summing to $O(Bn(d_c+d_d+1))$. Gradients, using the parameter-shift rule, experience a cost of $O(BL(3n-1))$, along with $O(Bn(d_c+d_d+1))$ for decoding. Hence, the per-step complexity can be denoted as $O(Bn(L(3n-1)+N_{\text{shot}}+d_c+d_d+1))$. Over T steps, E episodes, and N_e epochs with the rollout buffer $\mathcal{T}$, the total complexity can be denoted as $O(EN_eTBn(L(3n-1)+N_{\text{shot}}+d_c+d_d+1))$.

IV. Numerical Results and Discussions

A. Simulation Settings

This subsection presents the settings of parameters for the implementation of the proposed solution and simulations. The parameters of the QEDDPG and QEPPO algorithms are as follows. For QEPPO, the actor learning rate is set to 2×10^{-4}, the value learning rate to 2×10^{-3}, the discount factor γ to 0.99, the clipping parameter ϵ to 0.2, and the number of epochs N_e to 10 with a batch size $B=64$ [36]. For QEDDPG, the actor learning rate is 2×10^{-4}, the critic learning rate is 2×10^{-3}, the discount factor γ is 0.99, the soft update coefficient τ is 0.005, and the batch size is $B=64$. The training process is carried over $E=1000$ episodes for QEPPO and $E=1000$ episodes for QEDDPG, each episode composed of $T=200$ steps. Both algorithms employ HOE with $n=6$ qubits. The quantum circuit comprises $L=4$ layers and uses $N_{\text{shot}}=1024$ measurement shots. We carried out simulations using the `TorchQuantum` library [44]. This enables the implementation of QNNs and HOE on classical hardware. The simulator emulates the behavior of current noisy intermediate-scale quantum (NISQ) devices. This allows realistic testing under limited qubit counts and shallow circuit depths. Moreover, the selected circuit parameters were kept consistent with the operational capabilities of existing superconducting and photonic processors. Thus, this study focuses on algorithmic feasibility and learning performance rather than direct hardware deployment, ensuring that the proposed framework can be adapted to practical NISQ systems . Furthermore, maritime 6G-enhanced SAGIN environment parameters are summarized in Table III.

TABLE III
Maritime 6G-enhanced SAGIN environment parameters [22], [23], [43].

Parameters	Value
Number of maritime URLLC users, M	10
Number of sub tasks, ℓ	2
Number of berths, B	5
Number of antennas at UCR, N	64
Number of antennas at BS, K	8
Number of antennas at satellite, Y	16
Harbor size, $x^{\max}$, $y^{\max}$	(2000, 2000) m
Distance to LEO satellite, r	400 km
UCR speed, v_u	10 m/s
BS-MEC computation power, $F_{\text{bs}}^{\max}$	4 GHz
Local computation power, $F_{\text{local}}^{\max}$	1 GHz
Satellite-MEC computation power, $F_{\text{sat}}^{\max}$	8 GHz
Energy coefficients, ξ_{loc}, ξ_{bs}, ξ_{sat}	1×10^{-27} Watt.s^3/cycle^3
Task data size, D_m	$[1\times10^6-2\times10^6]$ bits
Task complexity, C_m	$[2\times10^8-4\times10^8]$ cycles
Maximum transmit power of user m, $P_m^{\max}$	5W
System bandwidth, $\mathcal{B}$	20 MHz
Balancing parameter, w_1 and w_2	0.5

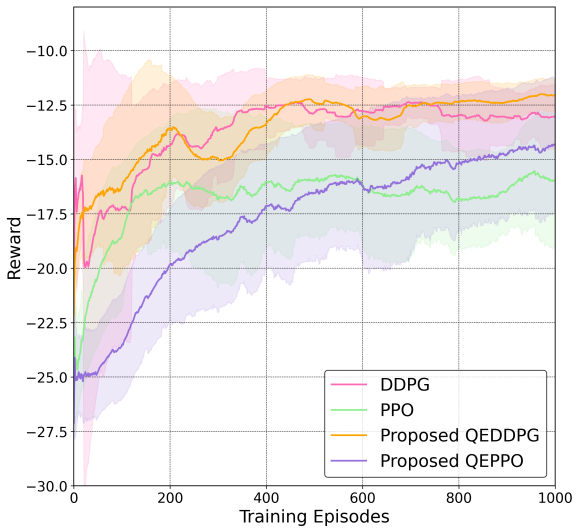

Fig. 3. Convergence performance.

B. Numerical Results

We comprehensively evaluate the effectiveness of the our proposed solutions by analyzing convergence performance and examining the impacts of key system parameters, including system bandwidth, task size, task complexity and URLLC user density.

1) Convergence performance: We evaluate the convergence behavior of the proposed QEDDPG and QEPPO frameworks in comparison with conventional DDPG and PPO algorithms, as shown in Fig. 3. All algorithms show an upward trend in reward as training continues, reflecting their ability to learn better policies over time. Among them, QEDDPG steadily improves and maintains a clear performance advantage across most of the training period. While DDPG initially shows fast progress, its learning becomes less consistent as training progresses, and it fails to sustain its early advantage. Furthermore, PPO starts with relatively high rewards, showing strong early learning behavior. However, as training progresses, the improvement slows down, and the model struggles to sustain its initial advantage. In contrast, QEPPO begins with a slower learning curve but steadily improves over time. Eventually, it surpasses PPO and maintains better performance in the later stages of training. Moreover, the proposed quantum-enhanced algorithms show better convergence and reward outcomes. QEDDPG consistently outperforms all others across the training process, while QEPPO demonstrates long-term reliability by steadily improving and overtaking its classical counterpart. These results shows the effectiveness of quantum-based learning in producing stable and efficient policy optimization compared to traditional reinforcement learning approaches.

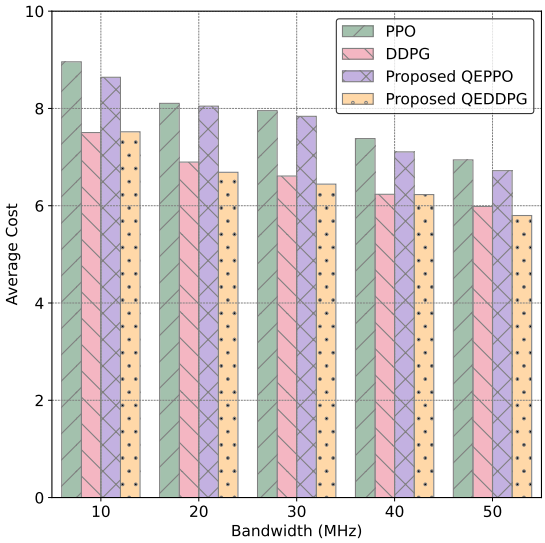

Fig. 4. Average cost with varying system bandwidth.

2) Average cost with varying system bandwidth: We analyze how varying the system bandwidth influences the average cost performance of the proposed QEDDPG and QEPPO frameworks relative to conventional DDPG and PPO algorithms, as shown in Fig. 4. As expected, an increase in system bandwidth generally leads to a decrease in the average cost across all methods, indicating more efficient resource utilization with greater available capacity. Initially, at lower bandwidth levels, all algorithms experience relatively higher costs. However, the proposed QEDDPG achieves the most efficient performance, maintaining a noticeable advantage over the other methods. QEPPO also demonstrates better cost efficiency than its conventional counterpart, PPO, even in the low bandwidth region. Moreover, as the bandwidth expands, all algorithms exhibit a clear reduction in cost. The advantage of the proposed methods remains evident, with QEDDPG consistently achieving the lowest cost across the entire bandwidth range. Although the conventional algorithms, particularly DDPG, show improvements as bandwidth increases, the performance gap between the proposed and conventional methods persists. Furthermore, at higher bandwidths, while all models continue to benefit from reduced costs, the gap between the quantum-enhanced and conventional methods narrows slightly. Despite this convergence, the proposed QEDDPG and QEPPO frameworks maintain superior cost efficiency compared to DDPG and PPO, validating the effectiveness of the quantum-enhanced learning approach under different system bandwidth conditions.

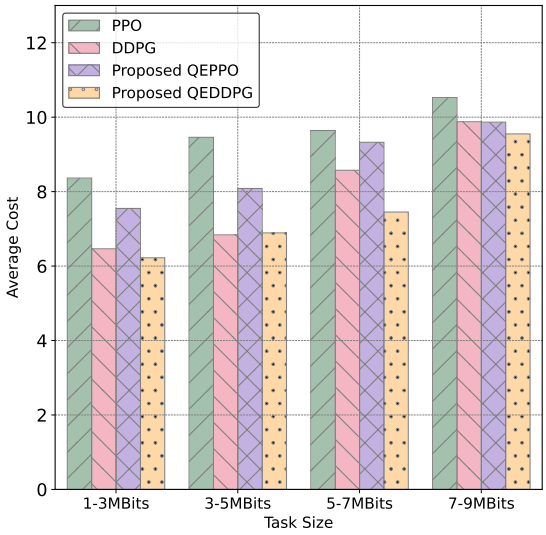

Fig. 5. Average cost with varying task size.

3) Average cost with varying task size: In this subsection, we evaluate how the average cost varies with task size. Fig. 5 presents the average cost measured across different task size ranges. As the task size increases, the average cost also rises for all algorithms, reflecting the growing resource demands associated with larger tasks. At smaller task sizes, all algorithms achieve relatively low costs, with the proposed QEDDPG demonstrating slightly better efficiency compared to the others. The proposed QEPPO also manages lower costs than conventional PPO, showing early benefits from the quantum-enhanced framework. As the task size becomes larger, the differences between the algorithms become more noticeable. The proposed QEDDPG consistently maintains a lower average cost compared to other methods, indicating more effective resource management under higher task demands. The proposed QEPPO also performs better than PPO, although the improvement is more moderate. Moreover, when the task sizes reach the higher ranges, all algorithms experience a noticeable increase in cost. However, the proposed quantum-enhanced

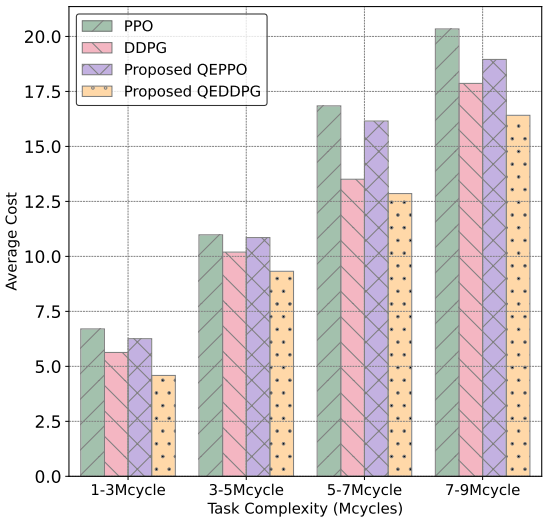

Fig. 6. Average cost with varying task complexity.

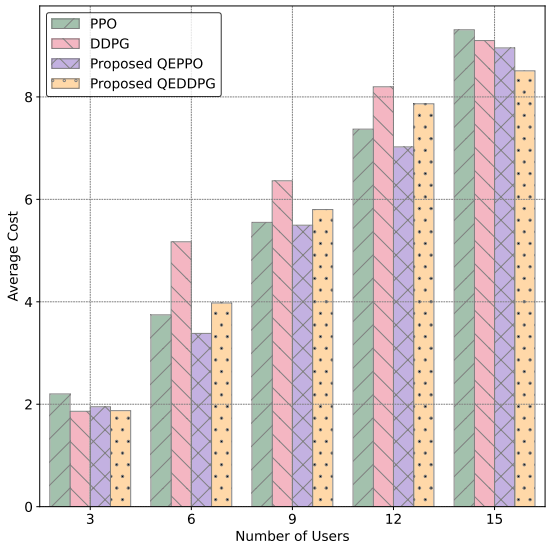

Fig. 7. Average cost with varying URLLC users.

methods, particularly QEDDPG, continue to deliver the best performance. These results confirm the ability of the quantum-enhanced frameworks to efficiently handle larger and more complex tasks.

4) Average cost with varying task complexities: In this subsection, we analyse the effect of task complexity on the average cost across all evaluated algorithms. As shown in Fig. 6, the average cost steadily increases as task complexity rises. This trend is expected because more complex tasks demand higher computational and communication resources, leading to greater system overhead. Moreover, the proposed QEDDPG framework consistently maintains the lowest average cost across the entire task complexity range, demonstrating strong efficiency even under increasing workload conditions. The proposed QEPPO also achieves better cost performance compared to the conventional PPO method, with the performance gap widening as task complexity becomes larger. Similarly, QEDDPG shows a clear advantage over DDPG, particularly at higher task complexities. These results highlight the ability of the proposed quantum-enhanced frameworks to manage complex resource allocation challenges more effectively than traditional DRL approaches.

5) Average cost with varying URLLC users: In order to investigate the effect of user density on system performance, we compare the average cost across different numbers of users in Fig. 7. As the number of users increases, the average cost rises across all methods, reflecting the growing resource demands associated with accommodating more users. At lower user counts, all algorithms achieve relatively low costs, and the differences between them are minimal. However, the proposed QEDDPG maintains a slight advantage by achieving the lowest average cost. The proposed QEPPO also shows improved cost management compared to PPO, indicating early benefits from the quantum-enhanced framework. Besides, as the number of users grows, performance differences among the algorithms become more noticeable. The proposed QEDDPG consistently achieves lower costs than the other methods, demonstrating superior resource allocation under increasing user loads. The proposed QEPPO also outperforms conventional PPO, although the improvement is more moderate compared to the gap between QEDDPG and DDPG. Furthermore, when the number of users reaches higher levels, all algorithms experience a noticeable rise in cost. However, both proposed quantum-enhanced models, particularly QEDDPG, continue to maintain the best performance. These results confirm the ability of the quantum-enhanced frameworks to manage higher user demands more efficiently than conventional approaches.

Thus, as shown in Fig. 3-7, DDPG and QEDDPG achieve lower average system cost and higher rewards than PPO and QEPPO in the 6G-enhanced SAGIN. This results from both algorithmic and implementation factors. DDPG is off-policy and uses a replay buffer, enabling extensive sample reuse and more critic updates per interaction. This is advantageous in our high-dimensional state–action space. In contrast, PPO is on-policy and discards old data, making it less sample-efficient in this setting. Moreover, our quantum feature map (HOE) and QNNs enrich the representation of the observation space with a different learning bias compared to classical networks. Due to quantum advantages, we observe improved critic fitting and actor conditioning. Both methods benefit from this improvement. However, QEDDPG, with its off-policy and deterministic strengths, converges faster and achieves lower average system cost.

V. Conclusion and Future Work

In this paper, we investigated a DT model with a maritime-enhanced SAGIN that addresses the growing need for intelligent resource management in dynamic harbor environments. By integrating MEC-enabled BS, UCR relays, and LEO satellites with MEC capabilities, the proposed system supports flexible and robust connectivity for dynamic URLLC users.

To effectively manage the complex task offloading, bandwidth allocation, local computation, and caching decisions under stringent latency and reliability requirements, we formulated a MINLP problem and solved it using a quantum-enhanced DRL frameworks namely QEDDPG and QEPPO. We used HOE to capture the nonlinear relationships of the observation space and proceed through the QNNs. According to the simulation results, the proposed quantum-enhanced DRL algorithms achieve lower system costs and superior resource allocation efficiency compared to conventional DRL baselines. The improvements are consistent across varying system conditions, demonstrating the resilience and adaptability of the proposed solutions under bandwidth expansion, increasing task sizes, and growing user densities. These results highlight the effectiveness of integrating quantum learning models into DT model with maritime-enhanced SAGIN systems for enabling scalable and adaptive communication networks.

To further advance the capabilities of the proposed framework, several promising research directions can be explored. One direction involves enrolling different quantum circuit architectures to improve the expressiveness and learning capacity of the policy and value networks. Another direction could examine alternative quantum encoding strategies beyond higher-order encoding to better capture complex state representations. Additionally, expanding the proposed framework toward multi-agent quantum reinforcement learning approaches could enable distributed optimization and coordination among terrestrial, aerial, and satellite nodes, further enhancing system scalability and efficiency in dynamic maritime environments.

REFERENCES

[1] Y. Liu, J. Yan, and X. Zhao, "Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4225–4236, April 2022.

[2] C. Zhuansun, P. Li, Y. Liu, and Z. Tian, "Generative AI-assisted mobile edge computation offloading in digital twin-enabled IIoT," *IEEE Internet Things J.*, May 2025.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] G. S. Kim, Y. Cho, S. Park, S. Jung, and J. Kim, "Quantum multi-agent reinforcement learning for joint cube-satellites and high-altitude long-endurance aerial vehicles in SAGIN," *IEEE Trans. Aerosp. Electron. Syst.*, Apr. 2025.

[9] M.-H. T. Nguyen *et al.*, "Real-time optimized clustering and caching for 6G satellite-UAV-terrestrial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 3009–3019, Mar. 2024.

[10] W. Mao, Y. Lu, G. Pan, and B. Ai, "UAV-assisted communications in SAGIN-ISAC: Mobile user tracking and robust beamforming," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 1, pp. 186–200, Jan. 2025.

[11] C. Lei, S. Wu, Y. Yang, J. Xue, and Q. Zhang, "Joint trajectory and communication optimization for heterogeneous vehicles in maritime SAR: Multi-agent reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12 328–12 344, Sep. 2024.

[12] 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.

[13] 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 Trans. on Tourism, Tech. and Intell.*, vol. 1, no. 1, pp. 1–8, Sep. 2024.

[14] K. K. Nguyen, S. R. Khosravirad, D. B. da Costa, L. D. Nguyen, and T. Q. Duong, "Reconfigurable intelligent surface-assisted multi-UAV networks: Efficient resource allocation with deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 358–367, Apr. 2022.

[15] M. Zhang, Z. Su, Q. Xu, Y. Qi, and D. Fang, "Energy-efficient task offloading in UAV-RIS-assisted mobile edge computing with NOMA," in *Proc. IEEE Int. Conf. on Computer Commun. workshops (INFOCOM workshops)*, Vancouver, Canada, May 20–23 2024, pp. 1–6.

[16] 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.

[17] J. Wu, M. Jia, Q. Guo, and X. Gu, "Efficient resource management based on DQN in LEO satellite edge computing system," in *Proc. IEEE Global Tele. Conf. Workshops (GLOBECOM Workshops)*, Kuala Lumpur, Malaysia, Dec. 4–8 2023, pp. 135–140.

[18] D. Wang, T. He, Y. Lou, L. Pang, Y. He, and H.-H. Chen, "Double-edge computation offloading for secure integrated space–air–aqua networks," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15 581–15 593, Sep. 2023.

[19] Y. Gao, Z. Ye, and H. Yu, "Cost-efficient computation offloading in SAGIN: A deep reinforcement learning and perception-aided approach," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3462–3476, Dec. 2024.

[20] K. Zheng, G. Jiang, X. Liu, K. Chi, X. Yao, and J. Liu, "DRL-based offloading for computation delay minimization in wireless-powered multi-access edge computing," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1755–1768, Mar. 2023.

[21] 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.

[22] 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.

[23] A. Paul, K. Singh, A. Kaushik, C.-P. Li, O. A. Dobre, M. D. Renzo, and T. Q. Duong, "Quantum-enhanced DRL optimization for DoA estimation and task offloading in ISAC systems," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 1, pp. 364–378, January 2025.

[24] 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.

[25] 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.

[26] 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.

[27] T. Yang, H. Feng, C. Yang, Y. Wang, J. Dong, and M. Xia, "Multivessel computation offloading in maritime mobile edge computing network," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4063–4073, Jun. 2019.

[28] M. Dai, N. Huang, Y. Wu, L. Qian, B. Lin, Z. Su, and R. Lu, "Latency minimization oriented hybrid offshore and aerial-based multi-access computation offloading for marine communication networks," *IEEE Trans. Commun.*, vol. 71, no. 11, pp. 6482–6498, Nov. 2023.

[29] Q. Guo, F. Tang, and N. Kato, "Routing for space–air–ground integrated network with GAN-powered deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 2, pp. 914–922, Apr. 2025.

[30] P. Zhang, C. Wang, N. Kumar, and L. Liu, "Space–air–ground integrated multi-domain network resource orchestration based on virtual network architecture: A DRL method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2798–2808, Mar. 2022.

[31] P. Zhang, Y. Li, N. Kumar, N. Chen, C.-H. Hsu, and A. Barnawi, "Distributed deep reinforcement learning assisted resource allocation algorithm for space–air–ground integrated networks," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3348–3358, Sep. 2023.

[32] F. Tang, C. Wen, L. Luo, M. Zhao, and N. Kato, "Blockchain-based trusted traffic offloading in space–air–ground integrated networks (SAGIN): A federated reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3501–3516, Dec. 2022.

[33] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, and M. Hangai, "A deep reinforcement learning-based dynamic traffic offloading in space–air–ground integrated networks (SAGIN)," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 276–289, Jan. 2022.

[34] Y. Xiao, Z. Ye, M. Wu, H. Li, M. Xiao, M.-S. Alouini, A. Al-Hourani, and S. Cioni, "Space–air–ground integrated wireless networks for 6G: Basics, key technologies, and future trends," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3327–3354, Dec. 2024.

[35] Y. Liu, Y. He, Y. Li, X. Wang, K. Zhang, M. Ju, Z. Ma, and Q. Tian, "Cost-oriented and delay-constrained anycasting for service function chain provisioning leveraging cloud–edge collaboration in space–air–ground integrated networks," *IEEE Internet Things J.*, vol. 12, no. 4, pp. 4475–4487, Feb. 2025.

[36] S. C. Prabhashana, D. V. Huynh, K. Singh, H.-J. Zepernick, O. A. Dobre, H. Shin, and T. Q. Duong, "Machine learning-based resource allocation in 6G integrated space and terrestrial networks-aided intelligent autonomous transportation," *IEEE Trans. Intell. Transp. Syst.*, Apr. 2025.

[37] C. Park, W. J. Yun, J. P. Kim, T. K. Rodrigues, S. Park, S. Jung, and J. Kim, "Quantum multiagent actor–critic networks for cooperative mobile access in multi-UAV systems," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 20 033–20 045, Nov. 2023.

[38] S. Park, C. Park, S. Jung, and J. Kim, "Adaptive quantum federated learning for autonomous surveillance multi-drone networks," *IEEE Trans. Intell. Veh.*, 2024.

[39] S. Park, G. S. Kim, and J. Kim, "Joint quantum reinforcement learning and neural myerson auction for high-quality digital-twin services in multi-tier networks," *IEEE Internet Things J.*, 2025.

[40] B. Narottama, T. Jamaluddin, and S. Y. Shin, "Quantum neural network with parallel training for wireless resource optimization," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5835–5847, May 2024.

[41] B. Narottama and S. Y. Shin, "Quantum neural networks for resource allocation in wireless communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1103–1115, Feb. 2022.

[42] Silvirianti, B. Narottama, and S. Y. Shin, "UAV coverage path planning with quantum-based recurrent deep deterministic policy gradient," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 7424–7429, May 2024.

[43] Y. Liu, J. Yan, and X. Zhao, "Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4225–4236, Apr. 2022.

[44] H. Wang, *et al.*, "Quantumnas: Noise-adaptive search for robust quantum circuits," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Seoul, South Korea, Feb. 2022.

[45] M. Monnet, N. Chaabani, T.-A. Drăgan, B. Schachtner, and J. M. Lorenz, "Understanding the effects of data encoding on quantum-classical convolutional neural networks," in *Proc. IEEE Int. Conf. on Quantum Comput. and Eng. (QCE)*, Montreal, QC, Canada, Sep. 15–20 2024.

[46] J. Wang, Y. Wang, P. Cheng, K. Yu, and W. Xiang, "DDPG-based joint resource management for latency minimization in NOMA-MEC networks," *IEEE Commun. Lett.*, vol. 27, no. 7, pp. 1814–1817, Jul. 2023.

[47] F. Chai, Q. Zhang, H. Yao, X. Xin, R. Gao, and M. Guizani, "Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7783–7795, Jun. 2023.