Quantum Reinforcement Learning for Lightweight LEO Satellite Routing

Gyu Seon Kim, Sungjoon Lee, In-Sop Cho, Soohyun Park, *Member, IEEE*, and Joongheon Kim, *Senior Member, IEEE*

***Abstract*—Low Earth orbit (LEO) satellite networks have emerged as a promising solution, offering advantages, such as lower propagation delay, broader coverage, and rapid deployment capabilities. However, the dynamic topology and frequent handovers inherent in LEO satellite systems, coupled with limited onboard computational resources, necessitate the development of efficient and lightweight routing algorithms. Therefore, this article proposes quantum reinforcement learning-based satellite routing (QRL-SR) tailored for LEO satellite networks. The QRL-SR algorithm addresses three critical considerations: 1) adapting to the dynamic and time-varying environment of LEO satellite networks; 2) incorporating LEO satellite geometry by transforming celestial coordinate data, specifically two-line element, into orbital coordinate systems for accurate LEO satellite positioning over time; and 3) being designed to be lightweight by leveraging QRL to reduce the number of training parameters. The proposed QRL-SR efficiently trains routing policies with fewer parameters, aligning with LEO satellites' small-size, weight, and power (SWaP) constraints. The primary purpose of the QRL-SR-based LEO satellites is to reduce free space path loss, delay time, and the number of hops needed for routing through the intersatellite links. Finally, experimental results demonstrate that the QRL-SR achieves routing performance comparable to or outperforms conventional algorithms while significantly reducing computational resources.**

***Index Terms*—Lightweight low Earth orbit (LEO) satellite routing algorithm, LEO satellite, quantum reinforcement learning (QRL).**

I. Introduction

DESPITE continuous advancements in the Internet and networking technologies, achieving seamless worldwide Internet connectivity remains a significant challenge. High-speed, uninterrupted Internet services are still lacking in many areas due to limited communication infrastructure, conflict zones, and natural disasters. Consequently, research on nonterrestrial networks (NTNs) as alternatives to traditional terrestrial networks has been actively pursued. The rise of 6G technology in recent years has further underscored the importance of NTNs. Among various NTN systems, low Earth orbit (LEO) satellites, which orbit at an altitude of approximately 500 km, have gained increasing attention as emerging components of NTNs [1]. Unlike traditional satellite communication systems that rely on geostationary (GEO) satellites, LEO satellites are favored in modern communication networks due to several advantages: 1) lower propagation delays due to their proximity to Earth; 2) broader coverage compared to other NTNs; 3) rapid and flexible global Internet service deployment; and 4) enhanced capacity and redundancy through large-scale LEO satellite constellations. However, LEO satellites also present challenges, such as 1) frequent handovers caused by their high-speed orbital motion at 7.5 km/s and 2) dynamic, time-varying network configurations. These factors must be carefully considered when designing and implementing algorithms for LEO satellite systems. One of the primary use cases for LEO satellite networks is the seamless global delivery of data from a source to its destination. Consequently, efficient routing algorithms are essential for LEO satellite constellation networks. The main goal of the LEO satellite routing algorithm proposed in this article is to reduce *the number of hops*, *free space path loss* and *delay time*. Thus, this article considers direct LEO satellite communication [2], [3], [4]. The intersatellite link (ISL) is a communication channel that allows LEO satellites to communicate directly with each other without involving the ground station (GS) [5]. ISL enables data to be routed through space from one LEO satellite to another, which can reduce the loss of the free space path, the delay time, and improve the resilience of the network. Several key aspects must be considered in the design of these algorithms. First, there is a need for *dynamic routing algorithm* that can cope with the dynamic and time-varying environment of LEO satellite networks [6]. LEO satellites orbit Earth based on their orbital elements, yet their relative positions to Earth change continuously over time due to the angular difference between the Earth's rotation axis and the orbital axis of the LEO satellite [7]. This leads to dynamic topological changes. Specifically, according to Kepler's third law, LEO satellites,

Received 17 April 2025; accepted 4 May 2025. Date of publication 9 May 2025; date of current version 9 July 2025. This work was supported in part by the Intelligent Technology Development Program on Disaster Response and Emergency Management funded by the Ministry of Interior and Safety (MOIS) under Grant 2022-MOIS37-005, and in part by the Institute for Information and Communications Technology Planning and Evaluation (IITP) Grant funded by Ministry of Science and ICT (MSIT) for Quantum AI Empowered Second-Life Platform Technology under Grant RS-2024-00439803 (SW Star Laboratory). *(Corresponding authors: In-Sop Cho; Soohyun Park; Joongheon Kim.)*

Gyu Seon Kim, Sungjoon Lee, and Joongheon Kim are with the Department of Electrical and Computer Engineering, Korea University, Seoul 02841, South Korea (e-mail: kingdom0545@korea.ac.kr; ssungjoon@korea.ac.kr; joongheon@korea.ac.kr).

In-Sop Cho is with the Satellite Communication Infra Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea (e-mail: lookatstar@etri.re.kr).

Soohyun Park is with the Division of Computer Science, Sookmyung Women's University, Seoul 04310, South Korea (e-mail: soohyun.park@sookmyung.ac.kr).

Digital Object Identifier 10.1109/JIOT.2025.3568454

which orbit at high speeds, experience frequent handovers, resulting in more dynamic topology updates than GEO and medium Earth orbit (MEO) satellites [8]. In other words, the characteristics of unique LEO satellite networks, such as dynamic topology and frequent handover, must be taken into account when designing the routing of LEO satellite constellations [9]. Second, *LEO satellite geometry*, i.e., the orbital coordinate systems, must be considered in the routing design. The raw information needed to calculate the orbit of the LEO satellites is based on the celestial coordinate system. However, it must be converted into the *orbital coordinate system* to calculate the latitude and longitude coordinate values of the LEO satellites that change over time. Based on these requirements, this article uses real LEO satellite data, i.e., two-line elements (TLE), based on the celestial coordinate system, to transform it into an orbital coordinate system through orbital dynamics [10]. This article constructs an experimental environment using *actual LEO satellite data* to increase the practical applicability of the proposed algorithm. Third, *lightweight LEO satellite routing training model* should be considered. LEO satellites have limited onboard resources [11]. LEO satellites are restricted by size, weight, and power (SWaP) limitations. The available computational power and memory of LEO satellites is significantly less than those of terrestrial network devices due to the need to minimize weight and energy consumption [12]. That is why it is crucial to design a lightweight LEO satellite routing algorithm that *reduces the number of training parameters*. A model with fewer parameters consumes less computational power and memory, aligning with the limited onboard resources of LEO satellites. This efficiency ensures that the LEO satellites can perform the necessary computations without overtaxing their hardware. Because quantum reinforcement learning (QRL) can design efficient LEO satellite routing algorithms considering all three of these considerations, this article proposes *quantum reinforcement learning-based satellite routing (QRL-SR)*, which allows lightweight LEO satellite routing. Conventional reinforcement learning (RL) has been a good solution in dynamic and uncertain environments, such as LEO satellite networks, as they continue to interact with the environment over time [8]. However, many training parameters are still required due to the use of the classical neural network (NN) [13]. Because QRL uses a quantum NN (QNN), unlike conventional RL, it can exponentially reduce the number of parameters required for training [14], [15], [16]. The QRL employs quantum superposition and entanglement to represent intricate state spaces using fewer quantum bits (qubits), thereby diminishing the number of necessary training parameters. In contrast to conventional RL, which is based on extensive classical NN, QRL efficiently encodes and processes large volumes of information through quantum parallelism. This capability enables the creation of lightweight LEO satellite routing algorithms that function effectively within LEO satellites' limited computational and energy resources. Consequently, QRL facilitates efficient policy training and real-time decision-making with fewer parameters, making it academically and industrially advantageous for LEO satellite constellation network applications. The QRL-SR algorithm reduces the number of training parameters exponentially while maintaining routing performance similar to that of QRL-SR. By evaluating the QRL-SR in a realistic experimental environment using TLE data, i.e., and actual orbital motion data for LEO satellites, this article improves its practical applicability within LEO satellite networks.

A. Contributions

The major contributions of the proposed QRL-SR can be summarized as follows.

1) *Minimizing Parameter Counts Through QRL-Based Routing Algorithm:* With available computational power and memory-limiting constraints, LEO satellites must reduce computational resources by reducing the number of training parameters. To achieve this objective, the lightweight LEO satellite routing algorithm is developed by minimizing training parameters through the implementation of the QRL-based approach that leverages the principles of quantum superposition and entanglement.
2) *High Applicability With Dynamic and Realistic Experimental Environments Using Real LEO Satellite Data:* The proposed algorithm is evaluated in an experimental environment based on TLE, the orbital data of real LEO satellites. The dynamic and realistic experimental environment designed on this basis increases the practical applicability of the QRL-SR algorithm proposed in this article to the real universe.

B. Organization

The remaining part of this article is organized as follows. Section II investigates the related work about LEO satellite routing problems. Section III introduces the foundations of quantum computing and QNN. In addition, Section IV describes the system models with orbit dynamic modeling of LEO satellites. Section V designs the proposed QNN update method. Moreover, Section VI evaluates the performance of the proposed QRL-SR algorithm. Lastly, Section VII concludes this article.

II. Preliminaries

A. Related Work

This section reviews the existing literature on satellite routing, routing strategies in the presence of satellite malfunctions, and quantum algorithm-based routing techniques. This review will provide context for our contributions to LEO satellite routing. Satellite routing has been a significant research area, particularly with the increasing deployment of LEO constellations [17], [18], [19]. Traditional satellite routing methods primarily focus on optimizing the link quality, minimizing latency, and ensuring high data throughput [20]. Additionally, with the rise, multilayer satellite network architectures have been proposed to enhance routing by integrating GEO and MEO satellites alongside LEO satellites to ensure global coverage and robustness [21]. Recent works have introduced the concept of ISL to improve data transfer efficiency across LEO constellations [22], [23]. This study has proposed a distributed satellite-terrestrial cooperative routing strategy based

on minimum hop-count analysis to optimize routing in mega LEO constellations [24]. Another study proposed a Logic Path Identified Hierarchical routing approach for large-scale LEO satellite networks, which partitions the constellation into multiple satellite groups and identifies logical paths to reduce complexity, enhance scalability, and improve routing convergence [25]. Routing strategies under satellite malfunctions have also garnered attention in recent years, motivated by the necessity for resilience in satellite networks. Malfunctions, such as communication failure or loss of a satellite, can significantly impact the overall network performance. Research efforts have been made to develop fault-tolerant routing protocols that can adapt to sudden satellite failures [26]. Deep RL (DRL) has been used to optimize routing in satellite networks, allowing adaptive decision-making in dynamic environments like existing malfunctioning satellites. The study proposed an intelligent routing algorithm for LEO satellites based on DRL, which adapts to satellite mobility to select optimal paths [27], [28]. Liu et al. [29] introduced a flow-centric DRL approach for high-throughput routing in LEO satellite broadband networks, demonstrating improved routing efficiency and throughput. QRL has emerged as a promising approach for routing like mobility control and industry, providing the ability to adaptively learn optimal intersatellite routing strategies in dynamic environments [30], [31]. This proposed method builds upon these existing efforts by proposing a novel QRL-based approach specifically tailored for lightweight LEO satellite routing. It incorporates the challenges of malfunctioning LEO satellite avoidance and efficient link utilization.

B. Algorithm Design Concepts of QRL-SR in LEO Satellite Systems Compared With Conventional Satellite Systems

In traditional GEO satellite networks, stable routing strategies are often based on conventional Internet protocol (IP) protocols, e.g., border gateway protocol (BGP) and multi protocol label switching (MPLS), which cater to the relatively unchanging topologies despite the inherent high propagation delays [32], [33]. MPLS guarantees quality of service (QoS) by forwarding data along predefined label switching paths, making it advantageous for managing delay and congestion [34]. When assuming stable physical links, as in GEO satellite networks, MPLS-based transmission paths offer relatively predictable performance [35]. GEO satellites rotate synchronously with the Earth, maintaining a fixed position relative to its surface. This synchronous rotation facilitates the establishment of satellite networks with static and unchanging topologies via the GEO platform. Similarly, MEO satellite networks, characterized by moderately dynamic link conditions, typically employ adaptive strategies, such as contact graph routing (CGR) and delay/disruption tolerant networking (DTN) protocols to handle intermittent connectivity and limited contact windows [36], [37], [38]. In contrast, LEO satellite networks operate under a fundamentally different set of constraints due to their lower orbital altitudes and high relative velocities, which induce rapid topological changes, frequent handovers, and highly variable link conditions [39]. As a consequence, performance metrics like hop count, free space path loss, and transmission delay become significantly more impactful in LEO environments; even marginal improvements in these parameters can lead to substantial overall performance gains. To address these unique challenges, our work proposes the QRL-based shortest path routing algorithm tailored specifically for LEO networks. This algorithm dynamically selects the nearest neighboring LEO satellite for packet forwarding, thereby minimizing the number of hops, reducing free space path loss, and lowering overall latency. By leveraging the adaptive and exploratory capabilities of QRL, the proposed approach is able to efficiently navigate the rapid connectivity fluctuations inherent in LEO systems, setting it apart from the more static and moderately adaptive routing protocols traditionally employed in GEO and MEO satellite networks.

III. Quantum Neural Network

A. Quantum Computing Basics

In the QNN, unlike classical NN, fundamental units other than bits are utilized for learning, significantly impacting how the system processes information. The qubit is the primary unit in quantum computing, which allows for unique properties, such as superposition, that distinguish it from classical bits. A classical system of n bits can be represented by 2^n possible states, each represented by a vector of length 2^n, with one nonzero entry indicating the specific state. Conversely, an n-qubit quantum state is a complex vector with 2^n dimensions, allowing it to simultaneously represent a superposition of multiple states.

In quantum mechanics, qubits can be represented in two basic states using bra-ket notation, which is expressed as

$$|0\rangle := \begin{bmatrix}1 & 0\end{bmatrix}^\top, |1\rangle := \begin{bmatrix}0 & 1\end{bmatrix}^\top. \tag{1}$$

Additionally, a single qubit state can be represented as a normalized 2-D complex vector, which can be expressed as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha & \beta\end{bmatrix}^\top \tag{2}$$

where α and β represent the complex probability amplitudes of the basis states $|0\rangle$ and $|1\rangle$, respectively, satisfying the condition $|\alpha|^2 + |\beta|^2 = 1$. Geometrically, a single qubit state can be visualized on the Bloch sphere, which is expressed as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{3}$$

where θ and ϕ represent angles that determine the position on the Bloch sphere, encapsulating the probability distribution of the state. For multiqubit systems, the quantum state $|\psi\rangle$ can be expressed in the Hilbert space as a superposition of basis states, which can be expressed as

$$|\psi\rangle = \sum_{l=0}^{2^q-1} \omega_l |l\rangle \tag{4}$$

where ω_l represents the probability amplitude of the lth basis state, and the sum of the squared amplitudes equals 1.

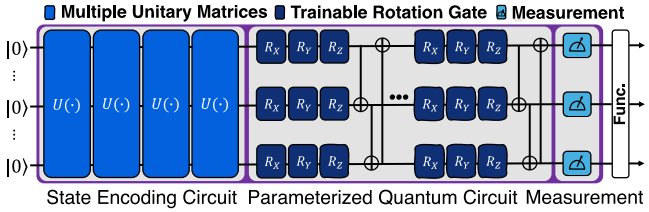

Fig. 1. Structure of the QNN.

B. Quantum Neural Network Structure

In classical NN, hidden layers apply linear and nonlinear transformations to the data, allowing the network to approximate complex functions effectively. Similarly, QNN performs transformations within the 3-D Bloch sphere, where quantum operations manipulate the quantum states of the qubits. These transformations are achieved using quantum gates, which perform linear and nonlinear operations on qubits, essential for learning intricate relationships in data. Linear transformations in QNN are implemented through unitary operations that preserve the norm of the quantum state. In contrast, nonlinear transformations often involve measurements and conditional operations that introduce nonlinearity, which is crucial for complex modeling. The integration of the RL with the QNN, i.e., QRL, enables the use of quantum advantages to develop sophisticated control mechanisms. As illustrated in Fig. 1, the QNN architecture can be divided into three main components: state encoding, parameterized quantum circuit (PQC), and measurement.

1) State Encoding: The state encoding component of the QNN transforms classical input data, denoted as χ_t, into a quantum state. Since quantum circuits operate on quantum states, this encoding is crucial to initialize the qubits appropriately. The encoder applies a series of unitary transformations, represented by $U_e(\chi_t)$, to map the classical data into a quantum state $|\psi_{0;t}\rangle$, which can be defined as

$$|\psi_{0;t}\rangle = U_e(\chi_t)\,|0\rangle^{\otimes q} \tag{5}$$

where the classical data χ_t dictate the parameters for the encoding gates. The encoding process is deterministic and does not involve trainable parameters, serving as the foundation for subsequent quantum processing.

2) Parameterized Quantum Circuit: The PQC component is analogous to the hidden layers in a classical NN. It comprises various quantum gates that transform the encoded quantum state. In QNN, Pauli, rotation, and controlled gates are used extensively [40]. Pauli gates, including X, Y, and Z, manipulate the quantum state by rotating it around specific axes of the Bloch sphere. Rotation gates $R_\Phi(\theta_k)$, parameterized by θ_k, allow for learnable transformations that adjust during training to optimize the network's output. Entanglement between qubits is established using controlled gates, such as the *Controlled*-Φ gate, which applies an operation to a target qubit conditioned on the state of a control qubit. The PQC applies a series of transformations to the quantum state, denoted as $|\psi_t\rangle$ at time t, which can be represented as

$$|\psi_t\rangle = \prod_{l=1}^{L} \boldsymbol{U}_l(\theta_t) U_e(\chi_t)\,|0\rangle^{\otimes q} \tag{6}$$

where $\boldsymbol{U}_l(\theta_t)$ represents the lth quantum layer containing trainable parameters. These parameters are optimized during training to improve the QNN's performance.

3) Measurement: After the quantum state passes through the PQC, it is measured to extract meaningful information. Measurement involves collapsing the quantum state to a classical state, which provides the observable properties of the system. Typically, measurements are performed along the z-axis of the Bloch sphere, but other axes may be used depending on the requirements. The measured observable is represented by the Hermitian matrix O. The expected value of measuring O given the quantum state $|\psi_t\rangle$ is expressed as

$$\langle\psi_t|\,O\,|\psi_t\rangle = \sum_{x=0}^{2^n-1} p(x|\theta_t) o_x \tag{7}$$

where $p(x|\theta_t)$ represents the probability of measuring outcome x, and o_x is the corresponding eigenvalue. This expectation value is used to evaluate and optimize the QNN during training, guiding the adjustment of the parameters θ_t to minimize the loss function.

The Number Parameters Reduction: In QNN, parameter reduction offers a significant advantage over classical NN. This is especially true in LEO satellite routing to support global network coverage. Classical NN typically requires many trainable parameters to accurately model complex functions, whereas QNN takes advantage of quantum properties, such as superposition and entanglement to represent these functions with fewer parameters [41]. Moreover, the relationship between channel count and parameter count in QNN is highly efficient: If the number of channels is $\mathcal{N}$, the parameters required in QNN increase linearly as $\mathcal{N} \times (3 \times 2 + 1)$, while in classical networks, the parameters tend to grow quadratically or cubically with the number of channels [15]. This efficiency is due to the design of each channel within the QNN. Specifically, each channel requires three rotation parameters (for rotations around the *RX*, *RY*, and *RZ* axes) to represent state information, and this structure is repeated across two layers to enhance the network's expressiveness. Additionally, each channel includes a bias parameter for fine-tuning the output, resulting in $3 \times 2 + 1 = 7$ parameters per channel. As a result, for $\mathcal{N}$ channels, the total parameter count scales linearly as $\mathcal{N} \times 7$, enabling QNN to handle complex tasks more efficiently than classical networks, where parameter count often grows nonlinearly with the channel count. In LEO satellite routing, where the task involves selecting the optimal next-satellite path based on latitude and longitude, the decision process can be distilled into four actions (moving in four directions based on the satellite's relative position). This results in four output channels in the QNN, representing each possible directional movement. Notably, while the number of channels corresponds to the action space, the number of qubits in the QNN corresponds to the size of the state space, which may include position and time-dependent data for each

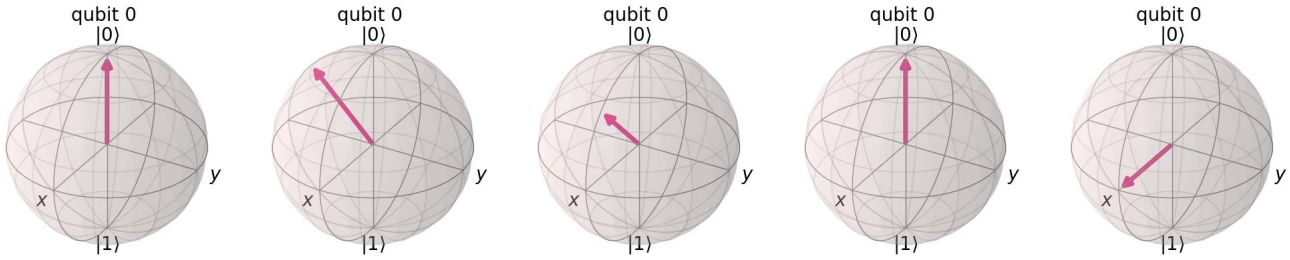

Fig. 2. Illustration of quantum states subjected to various gate operations on the Bloch sphere. (a) Original. (b) RX-gate ($\theta = [\pi/4]$). (c) RY-gate ($\theta = [\pi/4]$). (d) RZ-gate ($\theta = [\pi/4]$). (e) Hadamard-gate.

satellite. To manage this complexity, QNN is structured to accommodate both the number of channels and the state space requirements. For example, if a scenario requires $\mathcal{N}$ channels (or actions) and the state of each channel includes spatial and time-related data, the state space might expand to $\mathcal{N}^2$ (as in a 3-channel setup where nine distinct states are required). In such cases, three qubits (representing eight states) would not suffice and four qubits would be needed to fully represent the expanded state space. By efficiently leveraging the quantum state space, QNN achieves improved generalization with fewer trainable parameters, making it a promising solution for tasks where computational resources are constrained or overfitting is a concern. Note that the reduction in the number of training parameters does not entail any potential compromises. Compared to conventional RL, the reduced number of parameters in QRL represents an inherent advantage achieved by utilizing the QNN in place of the classical NN. Because the quantum circuit configuration of the QNN inherently requires fewer training parameters than the fully connected layer, this reduction does not come at the expense of routing performance, nor does it introduce any potential tradeoffs.

C. Quantum Gates

In our proposed QRL approach, the quantum circuit architecture, as depicted in Fig. 1 shows the overall architecture of the QNN, comprising a state encoding circuit, PQC, and measurement stages. The PQC includes RX, RY, and RZ gates, which are essential for encoding and optimizing the quantum state. Additionally, entanglement between qubits is facilitated by CNOT gates, as shown in the interconnections between qubits. Fig. 2 describes what happens when each Quantum Gate is taken from the Original State, as in Fig. 2(a).

RX-Gate: The RX gate in Fig. 2(b) is a rotation gate that acts on a single qubit, rotating it around the x-axis of the Bloch sphere. Mathematically, it is represented by the unitary operation: $RX(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$.

RY-Gate: The RY gate in Fig. 2(c) performs a rotation of the qubit state around the y-axis. It is represented by: $RY(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$.

RZ-Gate: The RZ gate in Fig. 2(d) rotates the qubit state around the z-axis. It is given by: $RZ(\theta) = \begin{bmatrix} e^{-i(\theta/2)} & 0 \\ 0 & e^{i(\theta/2)} \end{bmatrix}$.

Hadamard-Gate: The Hadamard gate in Fig. 2(e), often denoted as H, creates a superposition by transforming the qubit from a definite state (either $|0\rangle$ or $|1\rangle$) to a combination of both states. The Hadamard gate is represented by the following: $H = (1/\sqrt{2})\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

CNOT-Gate: The Controlled-NOT (CNOT) gate [42] which is describes $\bigoplus$ shape in Fig. 1 is a two-qubit gate that flips the target qubit if the control qubit is in the $|1\rangle$ state. The following matrix represents it: $\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. The CNOT gate introduces entanglement between qubits, a key feature in quantum computing. In this proposed QRL approach, the CNOT gate creates correlations between different qubits, enabling the model to learn complex relationships between routing paths in the satellite network.

IV. Dynamics Modeling of LEO Satellites for Robust Routing

A. Orbital Elements of LEO Satellites Processed Through TLE

To implement robust LEO satellite routing algorithms, it is necessary to locate LEO satellites expressed in latitude and longitude. TLE sets are required to calculate the latitude and longitude of LEO satellites that change over time. The TLE set has orbital data for real LEO satellites currently in orbit motion. TLE is a concise data format for specifying the orbital parameters of objects orbiting the Earth, such as LEO satellites. TLE is extensively used in LEO satellite tracking and modeling applications because it offers a standardized set of orbital elements that precisely define an object's position and trajectory in space at a specific epoch. The TLE set comprises two lines of textual data containing a series of numerical values that outline the orbital parameters of an LEO satellite relative to Earth. These parameters are expressed in a modified Keplerian format, which includes details, such as the orbit's inclination, right ascension of the ascending node (RAAN), eccentricity, argument of perigee, mean anomaly, and mean motion. Additionally, TLE provides information

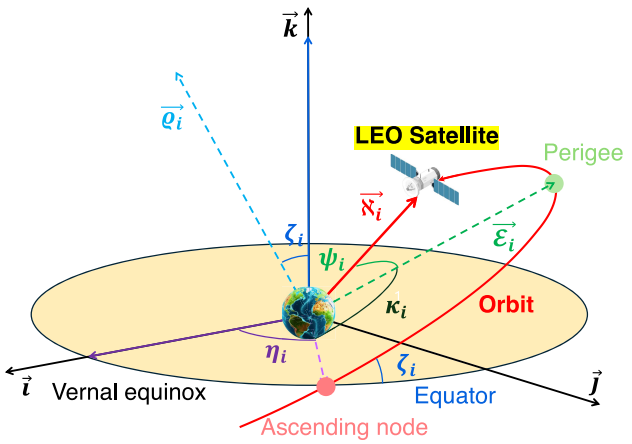

Fig. 3. Visual representation of satellite orbital elements.

like the epoch time, indicating the specific moment when the elements are valid, and a ballistic coefficient term that accounts for the effects of atmospheric drag over time. The TLE format adheres to the simplified general perturbations model 4 (SGP4) orbital model, which propagates the orbital elements to predict the LEO satellite's future positions. This method accounts for gravitational perturbations due to Earth's shape, solar radiation pressure, and atmospheric drag, allowing the TLE to provide relatively accurate predictions over short timeframes. However, TLE's accuracy degrades over time, necessitating periodic updates as external forces influence a satellite's orbit. These regular updates are made and provided by `Celestrak` [43] and `Space-Track` [44]. TLEs remain one of the most accessible methods for satellite tracking, providing essential information for collision avoidance, communication scheduling, and other mission-critical functions.

Each line within a TLE set encodes specific information in a streamlined format. The first line includes the LEO satellite's identifier, its classification, the epoch time, and the first and second derivatives of mean motion. In contrast, the orbital element data, e.g., orbital inclination, the RAAN, and eccentricity, etc., for predicting the future location of LEO satellites are located in the second line. The geometric shape of these orbital elements in terms of their orbits is shown in Fig. 3. The orbital elements used to locate LEO satellites are *inclination* (ζ), *RAAN* (η), *eccentricity* ($\mathcal{E}$), *argument of perigee* (κ), and *mean anomaly* ($\mathcal{M}$), which are in the second row of the TLE set. The *inclination* (ζ_i) is an orbital parameter within the TLE framework that defines the angle between the ith LEO satellite's orbital plane and the Earth's equatorial plane. It is measured in degrees, ranging from 0° to 180°, i.e., $\zeta_i \in [0°, 180°]$. An inclination of 0° signifies an equatorial orbit, while 90° indicates a polar orbit. Inclinations between 0° and 90° are classified as prograde, aligning with the Earth's rotation, whereas those between 90° and 180° are retrograde, opposing the Earth's rotation. The inclination significantly impacts an LEO satellite's coverage area, making it a critical factor in establishing mission objectives, determining satellite visibility, and identifying regions accessible for observation or communication. The *RAAN* (η_i) is an orbital parameter within the TLE framework that specifies the angle between the vernal equinox and the point where the ith LEO satellite crosses the equatorial plane from south to north, known as the *ascending node*. Measured in degrees [°], RAAN defines the orientation of the orbital plane in relation to the Earth's equatorial plane, with measurements ranging from 0° to 360°, i.e., $\eta_i \in [0°, 360°]$. The *eccentricity* ($|\vec{\mathcal{E}}_i| = \mathcal{E}_i$) defines the geometry of the ith LEO satellite's orbit around Earth. The eccentricity vector points ($\vec{\mathcal{E}}_i$) in the direction from the center of the orbit to the peripheries, i.e., point closest to Earth. It measures the extent of deviation from a perfectly circular orbit, with values ranging from 0 to less than 1, i.e., $\mathcal{E}_i \in [0, 1]$. An eccentricity of 0 indicates a circular orbit, while values approaching 1 signify increasingly elliptical orbits. Eccentricity helps determine how stretched or elongated the orbit is. Orbits with low eccentricity are nearly circular, resulting in relatively stable altitude and velocity, which are ideal for communication and Earth observation functions. Conversely, high-eccentricity orbits are more elliptical, causing significant variations in altitude and providing extended coverage over specific regions during different phases of the orbit. Eccentricity plays a crucial role in orbital dynamics, influencing an LEO satellite's velocity and the distance between it and Earth throughout its orbit. This parameter is essential for accurately calculating the satellite's position and velocity at any given time, which is expressed as $\mathcal{E}_i = \{1 - ([\mathfrak{a}_i^2/\mathfrak{b}_i^2])\}^{(1/2)}$, where $\mathfrak{a}_i$ and $\mathfrak{b}_i$ denote the semi-major axis and the semi-minor axis of the ith LEO satellite, respectively. The *argument of perigee* (κ_i) is an orbital parameter that specifies the orientation of the ith LEO satellite's elliptical orbit around Earth. It defines the angle between the ascending node and the perigee point, which is the closest location of the orbit to Earth. This angle is measured in the direction of the LEO satellite's motion and indicates the position of the perigee within the orbital plane. Typically expressed in degrees [°], the argument of perigee is essential for understanding the orbital geometry of an LEO satellite. It determines the orientation of the perigee relative to the ascending node, the point where the LEO satellite transitions from the southern to the northern hemisphere across the equatorial plane. This parameter is crucial as it influences the LEO satellite's altitude and position at various points in its orbit, thereby affecting ground coverage and the timing of observations. The *mean anomaly* ($\mathcal{M}_i(t)$) is an orbital parameter that signifies the position of the ith LEO satellite along its orbital path at a specific moment, known as the *epoch*. It is measured as an angle, typically in degrees [°], and represents the LEO satellite's progression around its orbit as though it were traveling at a constant average velocity. The mean anomaly does not correspond to the actual *true anomaly*, i.e., $\psi_i(t)$, which accurately indicates the ith LEO satellite's position relative to the central body. Instead, it is a theoretical value approximating the LEO satellite's position by assuming uniform motion along a *circular* reference orbit. Essentially, it functions as an estimation tool for determining the LEO satellite's location at a given time, making it a crucial component in propagating LEO satellite positions using models, such as the SGP4. The mean anomaly is used to calculate the eccentric anomaly, and true anomaly can be

obtained through the eccentric anomaly, which is expressed in (15) and (16). With knowledge of the mean anomaly, along with other parameters like eccentricity and eccentric anomaly, the LEO satellite's true position within its elliptical orbit can be predicted. This parameter is vital for tracking and forecasting the movements of Earth-orbiting objects, thereby assisting GSs and mission operators in maintaining accurate situational awareness of LEO satellite trajectories.

B. Latitude and Longitude Modeling of LEO Satellites

As mentioned above, data on LEO satellite orbits are expressed as TLE, which is raw data. In order to predict the exact location of the LEO satellite, the TLE data must be transformed into latitude and longitude via orbital dynamics equations [45]. To accurately track LEO satellite positions over time, these positions are delineated by *latitude* and *longitude* within the framework of orbital coordinate systems. Given that the raw data consists of coordinates within the *celestial coordinate systems*, it is imperative to convert celestial coordinates systems to *orbital coordinate systems* to determine latitude and longitude values. The latitude ($\mathcal{P}_i^{\phi}(t)$) and longitude ($\mathcal{P}_i^{\lambda}(t)$) of the ith LEO satellite can be expressed as

$$\mathcal{P}_i^{\phi}(t) = \sin^{-1}\left(\frac{\mathcal{P}_i(t)[3]}{\|\mathcal{P}_i(t)\|_{\mathcal{E}_2}}\right) \tag{8}$$

$$\mathcal{P}_i^{\lambda}(t) = \cos^{-1}\left(\frac{\mathcal{P}_i(t)[1]}{\|\mathcal{P}_i(t)\|_{\mathcal{E}_2}\cos\left(\mathcal{P}_i^{\phi}(t)\right)}\right) \tag{9}$$

where $\mathcal{P}_i(t)$ is a 3×1 matrix, which can be expressed as

$$\mathcal{P}_i(t) = \left[\rho_i^{\alpha} \times \rho_i^{\beta} \times \rho_i^{\gamma} \times \rho_i^{\delta}(t)\right] \times \sigma_i(t). \tag{10}$$

In (8) and (9), $\|\mathcal{P}_i(t)\|_{\mathcal{E}_2}$ signifies the *L2-norm*, i.e., *Euclidean norm*, of $\mathcal{P}_i(t)$, which can be expressed as, $\|\mathcal{P}_i(t)\|_{\mathcal{E}_2} = (\sum_{n=1}^{N} |\mathcal{P}_i(t)[n]|^2)^{(1/2)}$. Additionally, $\mathcal{P}_i(t)[1]$ and $\mathcal{P}_i(t)[3]$ refer to the first and third elements of $\mathcal{P}_i(t)$, respectively. In other words, *L2-norm* is defined as the square root of the sum of the squares of each vector component. In (10), ρ_i^{α}, ρ_i^{β}, ρ_i^{γ}, and $\rho_i^{\delta}(t)$ are coordinate transformation matrices that transform the coordinates of the ith LEO satellite from celestial coordinate systems to orbital coordinate systems, which can be expressed as (11), shown at the bottom of the page. In (11), η_i, ζ_i, and κ_i denote the RAAN, inclination, and argument of perigee of the ith LEO satellite, respectively. Furthermore, Θ_t in (11) denotes the integration of the Earth's rotational angular velocity over time t. This represents the angle by which the Earth rotates during the time interval t, which can be expressed as, $\int_{t=0}^{\tau} \varpi_E \Delta t$, where ϖ_E represents the Earth's rotational angular velocity. Additionally, the ith LEO satellite's coordinates calculated in celestial coordinate systems, i.e., $\sigma_i(t)$ in (10), can be expressed as

$$\sigma_i(t) = \begin{bmatrix} |\vec{\aleph_i}(t)|\cos(\psi_i(t)) \\ |\vec{\aleph_i}(t)|\sin(\psi_i(t)) \\ 0 \end{bmatrix} \tag{12}$$

where $\vec{\aleph_i}(t)$ and $\psi_i(t)$ denote the conic section vector and true anomaly of ith LEO satellite, respectively. Through $\aleph_i(t)$, the distance between the Earth's center and the ith LEO satellite and the coordinates of the ith LEO satellite are determined. In addition, $\aleph_i(t)$ is expressed as

$$|\vec{\aleph_i}(t)| = \frac{|\vec{\varrho}_i|^2}{\upsilon(1 + \mathcal{E}_i\cos(\psi_i(t)))} \tag{13}$$

where υ, $\mathcal{E}_i$ and $\vec{\varrho}_i$ denote the standard gravitational parameter of the Earth, the eccentricity, and the angular momentum vector of the ith LEO satellite, respectively. The magnitude of the angular momentum vector signifies the degree of conserved rotational momentum for an LEO satellite in orbit, contingent upon the absence of external torques. This magnitude is determined using the following formula:

$$|\vec{\varrho}_i| = \left[\upsilon\mathfrak{a}_i\left(1 - \mathcal{E}_i^2\right)\right]^{\frac{1}{2}} \tag{14}$$

where υ, $\mathfrak{a}_i$, and $\mathcal{E}_i$ denote the standard gravitational parameter, semi-major axis, and eccentricity of the ith LEO satellite, respectively. In addition, true anomaly of ith LEO satellite $\psi_i(t)$ in (12) and (13) is expressed as

$$\psi_i(t) = 2\tan^{-1}\left(\left[\frac{1 + \mathcal{E}_i}{1 - \mathcal{E}_i}\right]^{\frac{1}{2}}\tan\left(\frac{\Omega_i(t)}{2}\right)\right) \tag{15}$$

where $\Omega_i(t)$ denotes the eccentric anomaly of the ith LEO satellite. It can be expressed as

$$\Omega_i(t) = \mathcal{M}_i(t) + \mathcal{E}_i\sin\mathcal{M}_i(t) \tag{16}$$

where $\mathcal{M}_i(t)$ is the mean anomaly of the ith LEO satellite changes with time, which can be calculated as

$$\begin{aligned}\mathcal{M}_i(t) &= \mathcal{M}_{i_0} + \mathcal{H}_i \cdot \left(t - t_{i_0}\right) \\ &= \mathcal{M}_{i_0} + \left(\frac{\upsilon}{\mathfrak{a}_i^3}\right)^{\frac{1}{2}} \cdot \left(t - t_{i_0}\right)\end{aligned} \tag{17}$$

where t_{i_0} and $\mathcal{M}_{i_0}$ denote the reference time, i.e., the perihelion transit time of the ith LEO satellite and the mean anomaly of the ith LEO satellite at t_{i_0}, which can be obtained from the TLE set of the ith LEO satellite, respectively. In (17), $\mathcal{H}_i$ denotes the mean motion, which can be expressed as

$$\mathcal{H}_i = \frac{\upsilon}{\mathfrak{a}_i^3} = \frac{2\pi}{\mathfrak{T}_i} \tag{18}$$

where υ, $\mathfrak{a}_i$, and $\mathfrak{T}_i$ represent the standard gravitational parameter, the semi-major axis, and the orbital period of

$$\rho_i^{\alpha} = \begin{bmatrix} \cos(\eta_i) & \sin(\eta_i) & 0 \\ -\sin(\eta_i) & \cos(\eta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \rho_i^{\beta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\zeta_i) & \sin(\zeta_i) \\ 0 & -\sin(\zeta_i) & \cos(\zeta_i) \end{bmatrix}, \rho_i^{\gamma} = \begin{bmatrix} \cos(\kappa_i) & \sin(\kappa_i) & 0 \\ -\sin(\kappa_i) & \cos(\kappa_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \rho_i^{\delta}(t) = \begin{bmatrix} \cos(\Theta_t) & \sin(\Theta_t) & 0 \\ -\sin(\Theta_t) & \cos(\Theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

TABLE I
FUNDAMENTAL CONSTANTS IN LEO SATELLITE POSITIONING [46]

Constant	Value
Gravitational Constant, $\mathcal{Z}$	6.673 e-20
Mass of the Earth, $\mathcal{M}_e$	5.974 e+24 [kg]
Radius of the Earth, $\mathfrak{R}_e$	6.378 e+6 [m]
Standard gravitational parameter, $\upsilon = \mathcal{Z}\mathcal{M}_e$	3.986 e+14 [$m^3 \cdot s^{-2}$]
Earth's rotational angular velocity, ϖ_E	7.292 $\times 10^{-5}$ [rad/s]

the ith LEO satellite, respectively. The constants required to calculate the latitude and longitude of the ith LEO satellite are summarized in Table I, and the necessary variables for the computation are derived from the TLE set.

The distance between the ith and i'th satellites depends on each LEO satellite's changing latitude and longitude. Due to the Earth's nearly spherical shape, applying great-circle distance formulas to the longitude and latitude achieves a distance accuracy error of approximately 0.5% [47]. The distance between the ith and i'th satellites can be expressed as $\mathcal{D}_{i,i'}(t) = (\mathfrak{R}_e + \hbar_{i,i'}) \cdot \Delta\eth_{i,i'}(t)$, where $\mathcal{D}_{i,i'}(t)$, $\hbar_{i,i'}$, $\Delta\eth_{i,i'}(t)$ and $\mathfrak{R}_e$ denote the distance, average altitude, and center angle difference between the ith and i'th satellites and the radius of the Earth, respectively. The angle between each vector from the center of the Earth toward the ith and i'th satellites is defined as (19), shown at the bottom of the page In (19), $\Delta\mathcal{P}^{\phi}_{i,i'}(t)$ and $\Delta\mathcal{P}^{\lambda}_{i,i'}(t)$ are the latitude and longitude differences between the ith and i'th LEO satellites, which can be expressed as $\Delta\mathcal{P}^{\phi}_{i,i'}(t) = |\mathcal{P}^{\phi}_{i}(t) - \mathcal{P}^{\phi}_{i'}(t)|$ and $\Delta\mathcal{P}^{\lambda}_{i,i'}(t) = |\mathcal{P}^{\lambda}_{i}(t) - \mathcal{P}^{\lambda}_{i'}(t)|$, respectively. In addition, $\mathcal{P}^{\phi}_{i}(t)$, $\mathcal{P}^{\phi}_{i'}(t)$, and $\mathcal{P}^{\phi}_{m}(t)$ represent the latitudes of the ith/i'th LEO satellites and the average latitude of the ith/i'th LEO satellites, which can be expressed as, $\mathcal{P}^{\phi}_{m}(t) = (1/2)(\mathcal{P}^{\phi}_{i}(t)+\mathcal{P}^{\phi}_{i'}(t))$. In conclusion, the distance between the ith and i'th LEO satellites is expressed as

$$\mathcal{D}_{i,i'}(t) = 2\left(\mathfrak{R}_e + \hbar_{i,i'}\right) \cdot \left[\left(\sin\left(\frac{\Delta\mathcal{P}^{\lambda}_{i,i'}(t)}{2}\right) \cdot \cos\left(\mathcal{P}^{\phi}_{m}(t)\right)\right)^2 + \left(\cos\left(\frac{\Delta\mathcal{P}^{\lambda}_{i,i'}(t)}{2}\right) \cdot \sin\left(\frac{\Delta\mathcal{P}^{\phi}_{i,i'}(t)}{2}\right)\right)^2\right]. \quad (20)$$

This distance helps calculate the free space path loss in the reward function, expressed in (31).

V. ALGORITHM DESIGN

LEO satellite routing encompasses the strategies and protocols utilized to oversee data transmission pathways within a network of satellites orbiting the Earth at elevations ranging from approximately 500 km. These satellites travel at significant speeds relative to the Earth's surface and to one another, completing an orbit in approximately 90–120 min. Effective LEO satellite routing is essential for communication systems that depend on satellite constellations to deliver global coverage for Internet connectivity, telecommunications, and data relay services. The communication environment of LEO satellites is different from that of the terrestrial network and the existing GEO satellite-based communication environment. In contrast to GEO satellites, which maintain a fixed position relative to a specific point on Earth, LEO satellites are perpetually in motion. This continuous movement causes the network topology to change rapidly. Traditional routing protocols used in terrestrial networks and existing GEO satellite-based satellite networks are unsuitable for LEO satellite networks due to the dynamic topology. Handling frequent connection changes requires specialized protocols that can respond to these frequent environmental changes. The RL can exert great power in such a dynamic and uncertain environment. In particular, in satellite systems where light weight is essential, such as LEO satellites, the QRL with QNN reduces the number of parameters required for training, which is more beneficial to those systems [48], [49].

Our formulation is fundamentally rooted in QRL for scenarios involving the communication framework for lightweight LEO satellite routing. This article utilizes QRL in such multi-LEO satellite routing contexts and formalizes the problem using the Markov decision process (MDP). However, during the formalization with MDP, physical communication constraints prevent LEO satellites from directly observing all environmental states, making it more practical to employ a decentralized partially observable MDP (PO-MDP) [50], [51]. Because PO-MDP requires determining optimal actions based on incomplete information, solving these problems is generally more complex than fully observable MDP (FO-MDP). Nonetheless, PO-MDP offers a more *realistic* framework for scenarios involving multiple LEO satellites. In these situations, multiple LEO satellites operating under PO-MDP make sequential decisions based on partial environmental information. The reference model under consideration includes I LEO satellites. Each LEO satellite is represented as Ⓢ_i, i.e., ith LEO satellite, where $\forall \text{Ⓢ}_i \in \mathcal{I}$ and $|\mathcal{I}| = I$.

A. QRL Formulation for Lightweight LEO Satellite Routing

Fig. 4 illustrates an example of the proposed QRL-SR within LEO satellite constellation networks. In LEO satellite constellation networks, *source LEO satellite* (Ⓢ_S), *target LEO*

$$\Delta\eth_{i,i'}(t) = 2\left[\sin^2\left(\frac{\Delta\mathcal{P}^{\phi}_{i,i'}(t)}{2}\right) + \cos\left(\mathcal{P}^{\phi}_{i}(t)\right) \cdot \cos\left(\mathcal{P}^{\phi}_{i'}(t)\right) \cdot \sin^2\left(\frac{\Delta\mathcal{P}^{\lambda}_{i,i'}(t)}{2}\right)\right]^{\frac{1}{2}} = 2\left[\left(\sin\left(\frac{\Delta\mathcal{P}^{\lambda}_{i,i'}(t)}{2}\right) \cdot \cos\left(\mathcal{P}^{\phi}_{m}(t)\right)\right)^2 + \left(\cos\left(\frac{\Delta\mathcal{P}^{\lambda}_{i,i'}(t)}{2}\right) \cdot \sin\left(\frac{\Delta\mathcal{P}^{\phi}_{i,i'}(t)}{2}\right)\right)^2\right] \quad (19)$$

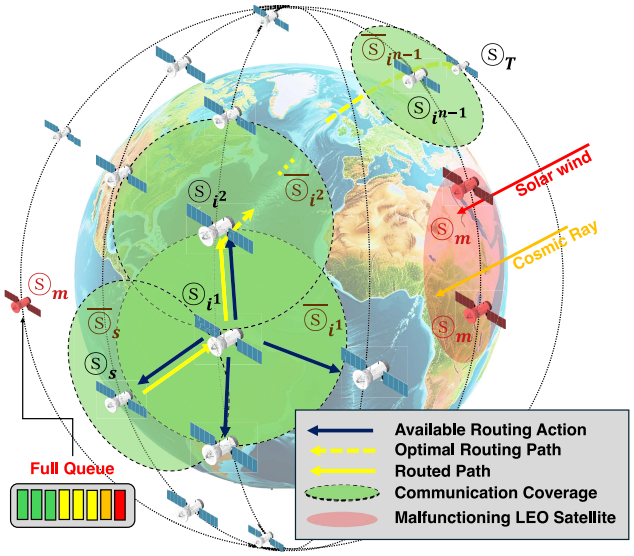

Fig. 4. Geographically close routing based on QNN related to network navigation.

satellite ($\textcircled{S}_T$), and *malfunctioning LEO satellites* ($\textcircled{S}_M$) are all part of the satellite set $\mathcal{I}$. In other words, they can be expressed as $\forall \textcircled{S}_S \in \mathcal{I} \quad \forall \textcircled{S}_T \in \mathcal{I} \quad \forall \textcircled{S}_M \in \mathcal{I}$, and $|\mathcal{I}| = I$. In addition, malfunctioning LEO satellites can be expressed as $\forall \textcircled{S}_M \in \mathcal{M}$ and $|\mathcal{M}| = \mathfrak{M}$, where the $\mathfrak{M}$ denotes the number of the malfunctioning LEO satellites. Malfunctioning LEO satellites denote those damaged by cosmic rays and solar winds or unable to be routed due to full queue backlogs. LEO satellites often fail due to cosmic rays and extreme temperature changes. LEO satellites that malfunction due to cosmic rays or drastic temperature fluctuations indicate link disruptions in the LEO network. Because data packets cannot be routed to malfunctioning LEO satellites, the links connected to them are considered broken. Furthermore, in the LEO satellite ISL routing environment, considering the queue backlog of each satellite is equivalent to taking into account the congestion within the LEO satellite network. As the network congestion increases, the queue backlog overhead for each satellite also rises, thereby reflecting the overall congestion in the LEO network. In the LEO satellite environment considered in this article, routing decisions are based on monitoring the queue backlog of each satellite, thereby inherently accounting for queuing delay in the network. As illustrated in Fig. 4, source satellites, target satellites, and malfunctioning satellites exist in LEO satellite constellation networks. Here, the source satellite with the current data packet has to transmit it to the target satellite through ISL, avoiding the malfunctioning satellites. The source LEO satellite calculates its distance from all the LEO satellites in satellite set $\mathcal{I}$ to route the data packets to the LEO satellites at the *smallest distance* to reduce the 1) *number of hops*; 2) *free space path loss*; and 3) *delay time*. In the LEO satellite environment considered in this article, ISL routing is performed by selecting the nearest LEO satellite to minimize delay time, which essentially indicates that QRL-SR accounts for latency. Moreover, because the ISLs in this article represent links between satellites, their latency is negligibly small compared to the latency experienced between the GS and the LEO satellite. In other words, the proposed QRL-SR algorithm initiates flow routing by transmitting data from the source LEO satellite to one of the closest satellites located within $\textcircled{S}_S$'s communication range, i.e., $\bar{\textcircled{S}}_S$. The i^1th satellite, i.e., $\textcircled{S}_{i^1}$, receiving the data packet from the source LEO satellite routes it to the nearest LEO satellite $\textcircled{S}_{i^2}$ within its communication range, i.e., $\bar{\textcircled{S}}_{i^1}$, to reduce free space path loss and delay time in the same way. Here, the superscript 2 of i in $\textcircled{S}_{i^2}$ means the second routed LEO satellite. In other words, the superscript n of i in $\textcircled{S}_{i^n}$ is the nth routed LEO satellite. At this point, the routing direction is from the source LEO satellite to the target LEO satellite. The i^{n-1}th satellite, i.e., $\textcircled{S}_{i^{n-1}}$, which has a data packet just before routing to the target LEO satellite, finally routes the data packet to the target LEO satellite when the target LEO satellite comes within the $\textcircled{S}_{i^{n-1}}$'s communication range, i.e., $\bar{\textcircled{S}}_{i^{n-1}}$, ending routing.

LEO satellites trained with QRL-SR follow the training procedure described above. To learn the routing strategy described above, QRL-SR initially requires iterative computational time during the training phase. However, once the training is complete, the real-time decision-making of LEO satellites during inference relies on matrix computations using pretrained parameters [52]. Therefore, even when deployed in large-scale satellite constellations with real-time constraints, QRL-SR demonstrates robust real-time routing performance through its use of pretrained parameters, thereby exhibiting strong scalability potential. Furthermore, because LEO satellites trained with QRL-SR continuously interact with a dynamically changing environment (a key characteristic of reinforcement learning), they can adapt to changes in network topology and handovers induced by satellite failures and high speed of LEO orbits.

1) State: The proposed QRL-SR algorithm routes to the nearest LEO satellite to reduce the number of hops, free space path loss and delay time. Therefore, the distance from each LEO satellite is the most critical factor, and to calculate this, the latitude and longitude of several LEO satellites enter the state of the QNN, i.e., the input layer of the QNN. The latitude and longitude coordinates of the I satellites that change over time are defined as states ($s \in \mathcal{S}$), which can be expressed as

$$\begin{aligned}\mathcal{S} \triangleq \Bigg\{ & \bigcup_{i\neq i'}^{I}\bigcup_{t=1}^{\mathcal{T}}\left\{\mathcal{P}_i^{\phi}(t)\right\}, \bigcup_{i\neq i'}^{I}\bigcup_{t=1}^{\mathcal{T}}\{\mathcal{P}_i^{\lambda}(t)\} \\ & \bigcup_{t=1}^{\mathcal{T}}\left\{\mathcal{P}_S^{\phi}(t)\right\}, \bigcup_{t=1}^{\mathcal{T}}\{\mathcal{P}_S^{\lambda}(t)\}, \bigcup_{t=1}^{\mathcal{T}}\left\{\mathcal{P}_T^{\phi}(t)\right\} \\ & \bigcup_{t=1}^{\mathcal{T}}\{\mathcal{P}_T^{\lambda}(t)\}, \bigcup_{i\neq i'}^{I}\bigcup_{t\in\mathcal{T}}\{\mathcal{D}_{i,i'}(t)\} \\ & \bigcup_{M\neq M'}^{\mathfrak{M}}\bigcup_{t=1}^{\mathcal{T}}\left\{\mathcal{P}_M^{\phi}(t)\right\}, \bigcup_{M\neq M'}^{\mathfrak{M}}\bigcup_{t=1}^{\mathcal{T}}\{\mathcal{P}_M^{\lambda}(t)\}\Bigg\} \end{aligned} \tag{21}$$

where $\mathcal{P}_i^{\phi}(t)$, $\mathcal{P}_i^{\lambda}(t)$, $\mathcal{P}_S^{\phi}(t)$, $\mathcal{P}_S^{\lambda}(t)$, $\mathcal{P}_T^{\phi}(t)$, $\mathcal{P}_T^{\lambda}(t)$, $\mathcal{P}_M^{\phi}(t)$, and $\mathcal{P}_M^{\lambda}(t)$ denote the set of latitude/longitude of the ith LEO satellite, source LEO satellite, target LEO satellite, and Mth malfunctioning LEO satellite, respectively. Lastly, the $\mathcal{D}_{i,i'}(t)$ represents the distance between the ith and i'th LEO satellites, which is expressed in (20). The latitude and longitude of the i-th LEO satellite and the latitude and longitude of the Mth LEO satellite can be expressed as

$$\mathcal{P}_i^{\phi}(t) \triangleq \{\phi_{t_1}^i, \phi_{t_2}^i, \ldots, \phi_{t_n}^i\}, \quad \bigcup_{n\neq n'}^{N} \{\phi_{t_n}^i\} \subseteq \{\mathcal{P}_i^{\phi}(t)\} \tag{22}$$

$$\mathcal{P}_i^{\lambda}(t) \triangleq \{\lambda_{t_1}^i, \lambda_{t_2}^i, \ldots, \phi_{t_n}^i\}, \quad \bigcup_{n\neq n'}^{N} \{\lambda_{t_n}^i\} \subseteq \{\mathcal{P}_i^{\lambda}(t)\} \tag{23}$$

$$\mathcal{P}_M^{\phi}(t) \triangleq \{\phi_{t_1}^M, \phi_{t_2}^M, \ldots, \phi_{t_n}^M\}, \quad \bigcup_{n\neq n'}^{N} \{\phi_{t_n}^M\} \subseteq \{\mathcal{P}_M^{\phi}(t)\} \tag{24}$$

$$\mathcal{P}_M^{\lambda}(t) \triangleq \{\lambda_{t_1}^M, \lambda_{t_2}^M, \ldots, \phi_{t_n}^M\}, \quad \bigcup_{n\neq n'}^{N} \{\lambda_{t_n}^M\} \subseteq \{\mathcal{P}_M^{\lambda}(t)\} \tag{25}$$

where $\phi_{t_n}^i$, $\lambda_{t_n}^i$, $\phi_{t_n}^M$, $\lambda_{t_n}^M$ denote the latitudes/longitudes of the ith and Mth malfunctioning LEO satellite at time t_n. Here, n, which represents the flow of time, has a constraint of $n \in (0, \infty)$, i.e., $N \triangleq \{n \in \mathbb{R}^+ | n > 0\}$. Physically, the flow of time is always continuous in the positive direction, but from the QRL perspective, QRL-SR-based satellites train in time steps. Thus, because n is a discrete variable (a natural number representing time) representing the flow of time, it can be expressed as, $N \triangleq \{n \in \mathbb{N} | n \geq 1\} \triangleq \{n \in \mathbb{Z}^+ | n \geq 1\}$. In this context, the latitude and longitude of the source and target LEO satellites are also expressed as

$$\mathcal{P}_S^{\phi}(t) \triangleq \{\phi_{t_1}^S, \phi_{t_2}^S, \ldots, \phi_{t_n}^S\}, \quad \bigcup_{n\neq n'}^{N} \{\phi_{t_n}^S\} \subseteq \{\mathcal{P}_S^{\phi}(t)\} \tag{26}$$

$$\mathcal{P}_S^{\lambda}(t) \triangleq \{\lambda_{t_1}^S, \lambda_{t_2}^S, \ldots, \phi_{t_n}^S\}, \quad \bigcup_{n\neq n'}^{N} \{\lambda_{t_n}^S\} \subseteq \{\mathcal{P}_S^{\lambda}(t)\} \tag{27}$$

$$\mathcal{P}_T^{\phi}(t) \triangleq \{\phi_{t_1}^T, \phi_{t_2}^T, \ldots, \phi_{t_n}^T\}, \quad \bigcup_{n\neq n'}^{N} \{\phi_{t_n}^T\} \subseteq \{\mathcal{P}_T^{\phi}(t)\} \tag{28}$$

$$\mathcal{P}_T^{\lambda}(t) \triangleq \{\lambda_{t_1}^T, \lambda_{t_2}^T, \ldots, \phi_{t_n}^T\}, \quad \bigcup_{n\neq n'}^{N} \{\lambda_{t_n}^T\} \subseteq \{\mathcal{P}_T^{\lambda}(t)\} \tag{29}$$

where $\phi_{t_n}^S$, $\lambda_{t_n}^S$, $\phi_{t_n}^T$, $\lambda_{t_n}^T$ represent the latitudes/longitudes of the source and target LEO satellites at time t_n. The coordinate values of all these LEO satellites are necessary to calculate the distance between the ith LEO satellite and the i' LEO satellite, i.e., $\mathcal{D}_{i,i'}(t)$, used in the reward function.

2) Action: As mentioned, the QRL-SR-based LEO satellite aims to reduce the number of hops, free space path loss and delay time by routing to the nearest LEO satellite. Thus, the action space, i.e., $a \in \mathcal{A}$, of QRL-SR is defined as

$$a^* = \arg \min_{i\in\mathcal{I},\, i\neq i'} \mathcal{D}_{i,i'}(t) = \arg \max_{i\in\mathcal{I},\, i\neq i'} \mathcal{R}(s, a, s'), \quad a^* \in \mathcal{A} \tag{30}$$

where $\mathcal{R}(s, a, s')$, a^*, and $\mathcal{A}$ denote the reward function, the optimal action, and the action space of the ith LEO satellite. The ith LEO satellite, which currently has a data packet, calculates the distance to all LEO satellites in set $\mathcal{I}$ except itself, i.e., $\mathbb{S}_i$, and routes it to the nearest LEO satellite [53]. It should be noted that routing the ith LEO satellite to the nearest LEO satellite means taking action to maximize the reward function value. This is because the training in QRL proceeds in the direction of maximizing the designed reward function value, and the reward function is defined in inverse proportion to the distance $\mathcal{D}_{i,i'}(t)$, as expressed in (31).

3) Reward: The LEO satellites receive reward values through the reward function $\mathcal{R}(s, a, s')$ each time the state transitions from the current state s to the subsequent state s' during environmental exploration. The purpose of LEO satellite routing based on QRL-SR proposed in this article is to reduce the 1) *number of hops*; 2) *free space path loss*; and 3) *delay time* by routing to nearby LEO satellites. Therefore, the reward function that the LEO satellite must maximize is designed as

$$\mathcal{R}(s, a, s') \triangleq -\left(\frac{4\pi \cdot \mathcal{D}_{i,i'}(t)}{\mho}\right)^2 \triangleq -\left(\frac{4\pi \cdot \mathcal{D}_{i,i'}(t) \cdot \wp}{\mathcal{C}}\right)^2 \tag{31}$$

where $\mho$, $\wp$, $\mathcal{C}$, and $\mathcal{D}_{i,i'}(t)$ denote the wavelength, the frequency, the velocity of light, and the distance between the ith and i'th LEO satellites, respectively. The communication band of ISL considered in this article is a *Ka*-band with a frequency range from 26.5 to 40 GHz and a wavelength range from 11.5 to 7.5 mm. The constants used in the experiment, including the channel condition, are summarized in Table II. The reward function defined in (31) is associated with free space path loss and can be reexpressed in dB form as follows:

$$\mathcal{R}(s, a, s') \triangleq -\{20\log(\wp) + 20\log(\mathcal{D}_{i,i'}(t)) + 92.44\}. \tag{32}$$

Based on the reward function defined in (31) and (32), the LEO satellite attempts to route to the nearest LEO satellite to maximize the reward function value. In addition, LEO satellites receive a minor negative reward, i.e., $\mathcal{U}$, every time step to route to the shortest path. When attempting to route to the malfunctioning LEO satellites, they also receive a negative reward, i.e., $\mathcal{Y}$. Lastly, to allow the LEO satellites to attempt to route toward the target LEO satellite, they receive the positive reward, i.e., ς, when they successfully route to the target LEO satellite.

In the QRL-SR-based ISL framework considered in this article, the probability of beam interference between links is low. The reasons for this are as follows. First, the ISL band considered in this article is the *Ka*-band; as shown in Table II, the ISL utilizes a high-frequency range, such as 30 GHz. On the LEO satellite, the spacing between multiple antennas is proportional to the wavelength and inversely proportional to the frequency. In other words, using high-frequency bands results in more compact antenna spacing on LEO satellites, which in turn enhances the directivity of the beams. An increase in beam directivity renders the probability of beam interference between links negligibly low. Second, in typical ISL scenarios, highly directional antennas with narrow beamwidths or optical links, e.g., laser communications, are

TABLE II
SYSTEM PARAMETERS AND HYPERPARAMETERS FOR PERFORMANCE EVALUATION

Notation	Value
Wavelength of the ISL ($\mho$)	10 mm
Frequency of the ISL ($\wp$)	30 GHz
Speed of light in vacuum ($\mathcal{C}$)	299,792,458 m/s
Discount factor (γ)	0.995
Batch size	64
Initial/Minimum of epsilon	0.70, 10^{-2}
Decaying epsilon	5×3.5^{-4}
Learning rate of the *actor* network (α_ϑ)	10^{-3}
Learning rate of the *critic* network (α_φ)	2.5×10^{-4}
Training episodes	40,000
Activation function	ReLU
Optimizer	Adam
Number of the quantum channels ($\mathcal{N}$)	4
Number of the qubits (q)	4

employed, and this article assumes the same. This outcome is primarily attributed to the employment of narrow-beam communication techniques and precise beamforming strategies, which enhance spatial isolation between individual links. Such designs concentrate energy in a specific direction, thereby significantly reducing the likelihood of spatial interference with adjacent links. Because intersatellite communications are conducted via narrow laser or RF beams, the probability of beam interference between links is exceedingly low. Third, as ISLs propagate through free space, unaffected by atmospheric conditions or obstructions, they inherently encounter far fewer multipath or reflection-based interference issues compared to terrestrial systems. This simplification of the communication environment naturally mitigates interference factors.

Although the extensive error analysis under varying channel conditions may be considered, it is essential to note that the proposed QRL-SR algorithm is designed for operation in high-frequency bands, i.e., *Ka*-band. Using such high-frequency bands provides inherently narrow beamwidth and high directivity, significantly mitigating interference and channel variability. The high-frequency band effectively minimizes interference effects, contributing to the overall resilience of the system. Therefore, the evaluation of proposed algorithm does not include detailed analysis under diverse channel conditions, as the system's operational environment ensures relatively stable channel characteristics. Instead, the robustness evaluation primarily focuses on other critical aspects, such as satellite malfunctions and network congestion.

B. Policy Training of QRL-SR-Based Lightweight LEO Satellites

Because QRL-SR-based LEO satellites train in the direction of maximizing the value of the reward function, the objective function of LEO satellites is defined as

$$\mathcal{J}(\vartheta) = \mathbb{E}_{\pi_\vartheta}\left[\sum_{t=1} \mathcal{R}\left(s, a, s'\right)\right] = \mathcal{V}_{\pi_\vartheta}(s_0) \tag{33}$$

where $\mathcal{J}(\vartheta)$, ϑ, $\mathcal{V}_{\pi_\vartheta}$, s_0, and π_ϑ denote the objective function, the training parameter, value function, start state, and the policy of the LEO satellites, respectively. The policy makes a probabilistic choice of one of several actions in a given state. This ensures that the choice of action is based on the probability distribution, which can be expressed as, $\pi_\vartheta(a, s) = \mathbb{P}(\mathcal{A}_t = a|\mathcal{S}_t = s)$. It means the probability of taking action a in state s at t. Optimal policy, i.e., π^*_ϑ, refers to LEO satellites' most effective action strategies to maximize cumulative reward in a given environment, which can be expressed as

$$\pi^*_\vartheta = \arg\max_\vartheta \mathbb{E}_{\pi_\vartheta}\left[\sum_{t=1}^{\tau} \gamma^{t-1} \cdot \mathcal{R}\left(s, a, s'\right)\right] \tag{34}$$

where γ represents the discount factor. The (33) can be re-expressed as

$$\begin{aligned}\mathcal{J}(\vartheta) &= \sum_{s\in\mathcal{S}} d(s) \cdot \mathcal{V}_{\pi_\vartheta}(s) \\ &= \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(a, s) \cdot \mathcal{R}\left(s, a, s'\right)\end{aligned} \tag{35}$$

where $\mathcal{V}_{\pi_\vartheta}(s)$ and $d(s)$ denote the state value function of the state s and the stationary distribution of the Markov chain for π_ϑ, respectively. In other words, $d(s)$ means the probability of starting training in state s, i.e., $s_0 \sim d(s)$. Here, the purpose of the LEO satellites is to maximize the objective function $\mathcal{J}(\vartheta)$ through the gradient accent method, which can be expressed as, $\vartheta' \leftarrow \vartheta + \alpha_\vartheta \cdot \nabla_\vartheta \mathcal{J}(\vartheta)$, where α_ϑ is the learning rate of the actor. The $\nabla_\vartheta \mathcal{J}(\vartheta)$ can be expressed as

$$\begin{aligned}\nabla_\vartheta \mathcal{J}(\vartheta) &= \nabla_\vartheta \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(a, s) \cdot \mathcal{R}\left(s, a, s'\right) \\ &= \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \nabla_\vartheta \pi_\vartheta(a, s) \cdot \mathcal{R}\left(s, a, s'\right) \\ &= \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(a, s) \cdot \frac{\nabla_\vartheta \pi_\vartheta(a, s)}{\pi_\vartheta(a, s)} \cdot \mathcal{R}\left(s, a, s'\right) \\ &= \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(a, s) \cdot \nabla_\vartheta \log \pi_\vartheta(a, s) \cdot \mathcal{R}\left(s, a, s'\right) \\ &= \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(a, s) \cdot \mathcal{R}\left(s, a, s'\right)\right] \\ &= \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(a, s) \cdot \mathcal{Q}_{\pi_\vartheta}(s, a)\right]\end{aligned} \tag{36}$$

where $\mathcal{Q}_{\pi_\vartheta}(s, a)$ denote the state-action value function, which can be expressed as

$$\mathcal{Q}_{\pi_\vartheta}(s, a) = \mathbb{E}_{\pi_\vartheta}[\mathcal{Z}_t|\mathcal{S}_t = s, \mathcal{A}_t = s] \tag{37}$$

where $\mathcal{Z}_t$ denote return at t. The return $\mathcal{Z}_t$ signifies the total accumulation of future weighted rewards expected to be received starting at time t, and it is defined as

$$\begin{aligned}\mathcal{Z}_t = \mathcal{R}\left(s, a, s'\right)_{t+1} + \gamma \mathcal{R}\left(s, a, s'\right)_{t+2} + \gamma^2 \mathcal{R}\left(s, a, s'\right)_{t+3} \\ + \gamma^3 \mathcal{R}\left(s, a, s'\right)_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k \mathcal{R}\left(s, a, s'\right)_{t+k+1}.\end{aligned} \tag{38}$$

In (38), the discount factor, i.e., γ, quantifies the present value of future rewards. It balances the tradeoff between short-term and long-term rewards, guiding the agent to prioritize immediate gains or to plan for future benefits. The discount factor typically takes a value between 0 and 1, i.e., $\gamma \in [0, 1]$. With the concept of the return and discount factor, the instantaneous reward, i.e., $\mathcal{R}(s, a, s')$, is replaced by a long-term value, i.e., $\mathcal{Q}_{\pi_\vartheta}(s, a)$. However, in optimizing the

policy of LEO satellites, it is more effective to use the advantage function, i.e., $\mathfrak{A}_{\pi_\vartheta}(s,a)$,than simply using the state-action value function, i.e., $\mathcal{Q}_{\pi_\vartheta}(s,a)$. The advantage function quantifies the relative benefit of taking a specific action a in a given state s compared to the average performance of all possible actions in that state under the current policy. The mathematical definition is expressed as follows. $\mathfrak{A}_{\pi_\vartheta}(s,a) = \mathcal{Q}_{\pi_\vartheta}(s,a) - \mathcal{V}_{\pi_\vartheta}(s)$. In other words, the advantage function essentially measures how much better (or worse) a particular action is compared to the policy's average action in that state. By deducting the baseline $\mathcal{V}_{\pi_\vartheta}(s)$ from the action-value function $\mathcal{Q}_{\pi_\vartheta}(s,a)$, the advantage function achieves a zero mean for updates. This adjustment markedly decreases the variance of the gradient estimates without introducing bias, thereby enhancing the stability of the learning process and facilitating more efficient policy updates. However, we need to check whether (36) holds even if we subtract $\mathcal{V}_{\pi_\vartheta}(s)$ from $\mathcal{Q}_{\pi_\vartheta}(s,a)$.

Lemma 1: That is, we need to check that the expected value of the original equation does not change even if $\mathcal{V}_{\pi_\vartheta}(s)$ is subtracted from (36). Thus, the following equation should be established:

$$\begin{aligned}&\mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{Q}_{\pi_\vartheta}(s,a)\right]\\&= \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \left\{\mathcal{Q}_{\pi_\vartheta}(s,a) - \mathcal{V}_{\pi_\vartheta}(s)\right\}\right]\\&= \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{Q}_{\pi_\vartheta}(s,a)\right]\\&\quad - \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)\right].\end{aligned}$$

Thus, the following conditions must be met,

$$\mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)\right] = 0. \tag{39}$$

Proof: Proof of formula (39). If the expectation operator of (39) is removed, it can be expressed as

$$\begin{aligned}&\mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)\right]\\&= \sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(s,a) \nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s).\end{aligned} \tag{40}$$

The right term, i.e., $\sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(s,a) \nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)$, in (40) can be expressed as follows:

$$\begin{aligned}&\sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(s,a) \nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)\\&= \sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \sum_{a\in\mathcal{A}} \pi_\vartheta(s,a) \frac{\nabla_\vartheta \pi_\vartheta(s,a)}{\pi_\vartheta(s,a)} \cdot \mathcal{V}_{\pi_\vartheta}(s)\end{aligned} \tag{41}$$

$$= \sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \sum_{a\in\mathcal{A}} \nabla_\vartheta \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)$$

$$= \sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \mathcal{V}_{\pi_\vartheta}(s) \sum_{a\in\mathcal{A}} \nabla_\vartheta \pi_\vartheta(s,a) \tag{42}$$

$$= \sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \mathcal{V}_{\pi_\vartheta}(s) \nabla_\vartheta \sum_{a\in\mathcal{A}} \pi_\vartheta(s,a)$$

$$= \sum_{s\in\mathcal{S}} d_{\pi_\vartheta}(s) \mathcal{V}_{\pi_\vartheta}(s) \nabla_\vartheta \cdot 1 = 0 \tag{43}$$

$$\therefore \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(s,a) \cdot \mathcal{V}_{\pi_\vartheta}(s)\right] = 0 \tag{44}$$

where $d_{\pi_\vartheta}(s)$ is the state distribution, which is a distribution representing the rate at which LEO satellites acting according to policy π_ϑ stay in each state s on average. In (41), $\nabla_\vartheta \log \pi_\vartheta(s,a)$ can be expressed as, $\nabla_\vartheta \log \pi_\vartheta(s,a) = (\nabla_\vartheta \pi_\vartheta(s,a)/\pi_\vartheta(s,a))$. In (42), $\mathcal{V}_{\pi_\vartheta}(s)$ is the state value function that can escape outside the sigma because it is the value related to state s and not to action a. Additionally, a critical property of policy is that, for each state, the sum of the probabilities of all possible actions equals 1. Thus, in (43), the sum of the policy is 1, i.e., $\sum_{a\in\mathcal{A}} \pi_\vartheta(s,a) = 1$. Here, because the constant of 1 is differentiated, it becomes 0, and in conclusion, (44) can be obtained. This concludes the proof for (39). ■

Therefore, the (36) can be expressed in an alternative form as

$$\nabla_\vartheta \mathcal{J}(\vartheta) = \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(a,s) \cdot \mathfrak{A}_{\pi_\vartheta}(s,a)\right]. \tag{45}$$

In (45), it can be changed to δ_{π_ϑ}, a temporal difference (TD) error instead of $\mathfrak{A}_{\pi_\vartheta}(s,a)$. The TD error can be expressed as

$$\delta_{\pi_\vartheta} = \underbrace{\mathcal{R}(s,a,s') + \gamma \mathcal{V}_{\pi_\vartheta}(s')}_{\text{TD Target}} - \mathcal{V}_{\pi_\vartheta}(s) \tag{46}$$

where *TD target* constitutes the accurate solution derived from the Bellman optimality equation and is defined as the sum of the immediate reward at the current time step and the discounted cumulative reward from the subsequent time step. The expected value of δ_{π_ϑ} is calculated as

$$\begin{aligned}\mathbb{E}_{\pi_\vartheta}\left[\delta_{\pi_\vartheta}|s,a\right] &= \mathbb{E}_{\pi_\vartheta}\left[\mathcal{R}(s,a,s') + \gamma \mathcal{V}_{\pi_\vartheta}(s') - \mathcal{V}_{\pi_\vartheta}(s)\right]\\&= \mathbb{E}_{\pi_\vartheta}\left[\mathcal{R}(s,a,s') + \gamma \mathcal{V}_{\pi_\vartheta}(s')|s,a\right] - \mathcal{V}_{\pi_\vartheta}(s)\\&= \mathcal{Q}_{\pi_\vartheta}(s,a) - \mathcal{V}_{\pi_\vartheta}(s) = \mathfrak{A}_{\pi_\vartheta}(s,a).\end{aligned} \tag{47}$$

Based on (47), the expected value of the TD error, i.e., δ_{π_ϑ}, is the advantage function, i.e., $\mathfrak{A}_{\pi_\vartheta}(s,a)$. In other words, the TD error is an *unbiased estimate* of the advantage function. This means that when the LEO satellite obtains TD errors in the environment and averages them, the average value converges to the advantage function. The advantage function of (45) is replaced by delta, which means that it is okay to update the learning using the TD target. Consequently, the objective function of the LEO satellites can finally be expressed as

$$\nabla_\vartheta \mathcal{J}(\vartheta) = \mathbb{E}_{\pi_\vartheta}\left[\nabla_\vartheta \log \pi_\vartheta(a,s) \cdot \delta_{\pi_\vartheta}\right]. \tag{48}$$

The LEO satellites, i.e., actor, learn to maximize the objective function in a gradient ascent method, and the updated expression of the actor network parameter, i.e., ϑ, is expressed as

$$\vartheta \approx \vartheta + \alpha_\vartheta \times \left[\nabla_\vartheta \log \pi_\vartheta(a,s) \cdot \delta_\varphi\right] \tag{49}$$

where φ is the network parameter of the critic network. The δ_φ denote the TD error parameterized by φ, which can be expressed as

$$\delta_\varphi = \mathcal{R}(s,a,s') + \gamma \mathcal{V}_\varphi(s') - \mathcal{V}_\varphi(s) \tag{50}$$

and then, the state value function parameterized by φ is expressed as

$$\mathcal{V}_\varphi(s) = \mathbb{E}_{\pi_\vartheta}\left[\sum_{u=t}^{\tau} \gamma^{u-t} \cdot \mathcal{R}(s,a,s')\right]. \tag{51}$$

The state value function, i.e., $\mathcal{V}_{\pi_\vartheta}(s)$, is approximated as state value function parameterized by φ, i.e., $\mathcal{V}_\varphi(s)$, to convert it into

a training algorithm through the QNN of the critic network. Because the actual value function is unknown, the QNN-based critic network is used to approximate it. At this point, the loss function gradient of the critic network, which evaluates how good the action taken by the actors, i.e., LEO satellites, is defined as

$$\nabla_{\varphi}\mathcal{L}(\varphi)=\sum_{t=1}^{\tau}\nabla_{\varphi}\left\|\delta_{\varphi}\right\|^{2} \tag{52}$$

The critic network optimizes its parameters φ to *minimize* the loss function, defined in (52). In other words, the purpose of the critic network is to reduce TD error. In this manner, TD-based actor-critic methods update previous estimates using future estimates [54]. To minimize the discrepancy between the current state value function, i.e., $\mathcal{V}_{\varphi}(\cdot)$, computed by the critic network, and the TD target, the parameters of the critic network are adjusted to reduce the TD error, i.e., $\delta\varphi$, through gradient descent method. This optimization process can be expressed as

$$\varphi\approx\varphi+\alpha_{\varphi}\times\left[\,\delta_{\varphi}\cdot\nabla_{\varphi}\mathcal{V}_{\varphi}(s)\,\right] \tag{53}$$

where α_{φ} denotes the learning rate of the critic network parameterized by φ, a hyperparameter that specifies the magnitude of learning updates applied at each step. In this way, the QRL-SR algorithm is implemented through two networks, the actor and critic network, parameterized by ϑ and φ, respectively. To optimize the objective functions of the actor network and minimize the loss function of the critic network, the derivatives with respect to the ℓth parameters of both the actor and the critic are defined as

$$\frac{\partial\mathcal{J}(\vartheta)}{\partial\vartheta_{\ell}}=\underbrace{\frac{\partial\mathcal{J}(\vartheta)}{\partial\pi_{\vartheta}}\cdot\frac{\partial\pi_{\vartheta}}{\partial\langle\mathcal{O}_{\ell,\vartheta}\rangle}}_{\text{Classical Backpropagation}}\cdot\underbrace{\frac{\partial\langle\mathcal{O}_{\ell,\vartheta}\rangle}{\partial\vartheta_{\ell}}}_{\text{Parameter Shift Rule}} \tag{54}$$

$$\frac{\partial\mathcal{L}(\varphi)}{\partial\varphi_{\ell}}=\underbrace{\frac{\partial\mathcal{L}(\varphi)}{\partial\mathcal{V}_{\varphi}(s)}\cdot\frac{\partial\mathcal{V}_{\varphi}(s)}{\partial\langle\mathcal{O}_{\ell,\varphi}\rangle}}_{\text{Classical Backpropagation}}\cdot\underbrace{\frac{\partial\langle\mathcal{O}_{\ell,\varphi}\rangle}{\partial\varphi_{\ell}}}_{\text{Parameter Shift Rule}} \tag{55}$$

where ϑ_{ℓ} and ϑ_{ℓ} denote the ℓth parameter of the actor and critic networks, respectively. In (54) and (55), the $\langle\mathcal{O}_{\ell,\vartheta}\rangle$ and $\langle\mathcal{O}_{\ell,\varphi}\rangle$ represent the *observable* corresponding to the ℓth basis of the actor and critic networks, respectively. The initial and secondary derivatives of both right-hand sides are calculated using classical partial differentiation, i.e., classical backpropagation. However, the subsequent tertiary derivatives cannot be determined through classical methods due to the indeterminate nature of the quantum state prior to its collapse via *measurement*. To address this issue, the parameter shift rule (PSR) is utilized for parameter optimization throughout the training process [55]. When applied to the derivative of the actor's ℓth parameter with respect to the 0th derivative, this rule is expressed as

$$\frac{\partial\langle\mathcal{O}_{\ell,\vartheta}\rangle}{\partial\vartheta_{\ell}}=\langle\mathcal{O}_{\ell,\vartheta+\frac{\pi}{2}\mathbf{e}_{\ell}}\rangle-\langle\mathcal{O}_{\ell,\vartheta-\frac{\pi}{2}\mathbf{e}_{\ell}}\rangle \tag{56}$$

where $\mathbf{e}_{\ell}$ represents the ℓth basis vector. In contrast to backpropagation, the *PSR* provides a more direct and straightforward method. Consequently, training within QNN can be expedited [49].

C. Computational Complexity Analysis of QRL-SR

This section mathematically analyzes the computational demand of QRL-SR, which is composed of the quantum actor and the quantum critic. The computational demand per episode for QRL-SR can be expressed as the sum of the computational demand required to compute the TD error in (50) and the computational demand required to evaluate the objective function in (48). This computational process can be mathematically expressed as [30]

$$\begin{aligned}F_{\text{QRL-SR}}&=F^{TD}_{\text{QRL-SR}}+F^{\text{Obj}}_{\text{QRL-SR}}\\&=\Im\Big(\tau\cdot\Big(F^{CN}_{\text{QRL-SR}}+|\mathcal{A}(t)|+F^{AN}_{\text{QRL-SR}}\Big)\Big).\end{aligned} \tag{57}$$

where $F^{TD}_{\text{QRL-SR}}$ and $F^{\text{Obj}}_{\text{QRL-SR}}$ denote the computational demands for TD error and objective function, respectively. The computational demand for TD error, i.e., $F^{TD}_{\text{QRL-SR}}$, can be expressed as

$$F^{TD}_{\text{QRL-SR}}=2\times\tau\times F^{CN}_{\text{QRL-SR}} \tag{58}$$

where $F^{CN}_{\text{QRL-SR}}$ represents the computational demand of the quantum critic network, which can be expressed as

$$F^{CN}_{\text{QRL-SR}}=\Im(|\mathcal{S}|+|\varphi|+\mathfrak{O}). \tag{59}$$

In (59), $|\mathcal{S}|$, $|\varphi|$, and $|\mathfrak{O}|$ stand for the input dimension, number of parameters, and output dimension of the critic NN, respectively. Because the quantum critic NN in QRL-SR is responsible for computing the state value function corresponding to the actions taken by the quantum actor NN, its output dimension is one, i.e., $|\mathfrak{O}|=1$. In (57), the computational demand for evaluating the objective function can be expressed as

$$F^{\text{Obj}}_{\text{QRL-SR}}=2\times\tau\times\left(|\mathcal{A}|+F^{AN}_{\text{QRL-SR}}\right) \tag{60}$$

where $F^{AN}_{\text{QRL-SR}}$ signifies computational demand of the quantum actor network, which may be represented as

$$F^{AN}_{\text{QRL-SR}}=\Im(|\mathcal{S}|+|\vartheta|+|\mathcal{A}|) \tag{61}$$

where $|\vartheta|$ and $|\mathcal{A}|$ denote the number of parameters and action dimension of the actor NN, respectively. The state encoder entails an operation count proportional to $\mathcal{S}$, while the parameterized quantum gate operations necessitate an operation count proportional to $|\vartheta|$. Moreover, $|\mathcal{A}|$ quantum measurements and softmax computations are required to generate the action outputs. By contrast, the computational demand of conventional RL methods employing classical NN is expressed as the sum of the computational requirements for the actor and critic networks. Mathematically, this can be represented as

$$\begin{aligned}F_{\text{CRL}}&=F^{CN}_{\text{CRL}}+F^{AN}_{\text{CRL}}\\&=\Im(\tau\times(|\mathcal{S}|\times|\vartheta|\times|\mathcal{A}|+|\mathcal{S}|\times|\varphi|))\end{aligned} \tag{62}$$

where F^{AN}_{CRL} and F^{CN}_{CRL} stand for the computational demands of the actor and critic networks in conventional RL. The computational demand of the actor NN can be expressed as

$$F^{AN}_{\text{CRL}}=\Im(|\mathcal{S}|\times|\vartheta|\times|\mathcal{A}|). \tag{63}$$

Then, the computational demand of the critic NN can be expressed as

$$F_{\mathrm{CRL}}^{CN} = \eth(|\mathcal{S}| \times |\varphi|). \tag{64}$$

These equations demonstrate that the computational demand of the QRL-SR is lower than that of the conventional RL approaches using classical NN. The fact that QRL-SR's computational demand is lower than that of conventional RL approaches underscores its potential for implementation in LEO satellite network environments, where computing resources and memory are limited.

VI. Performance Evaluation

A. Simulation Setup

The performance of the proposed QRL-SR algorithm undergoes rigorous evaluation using real-world LEO satellite constellation data, specifically TLE data obtained from `Celestrak` [43] and `Space-Track` [44]. The TLE data sourced from `Celestrak` and `Space-Track` are converted from orbital elements into the LEO satellites' latitudes and longitudes, which vary over time, using (8)–(18). These coordinates serve as the basis for calculating the distances between inter-LEO satellites as defined in (20). The system parameters and hyperparameters employed in the experiment are detailed in Table II.

The LEO satellites equipped with the proposed QRL-SR algorithm in this article are fundamentally based on quantum computing. In general, quantum computing technologies typically require extremely low temperatures or stringent external conditions, necessitating the use of large-scale cooling equipment and complex external control systems to maintain a cryogenic environment. However, when quantum computers are deployed in LEO satellite systems, the natural cooling provided by the space environment eliminates the need for such elaborate cooling apparatus, thereby offering a distinct advantage. Due to these characteristics, quantum computers exhibit considerable potential for widespread application in aerospace systems [14], [56]. Furthermore, as summarized in Table II, this article employs just four qubits, which is considered an acceptable level given the current advancements in quantum computing and satellite systems [57]. Because the processing power of quantum computers generally depends on the number of qubits, the proposed QRL-SR can be efficiently operated in resource-constrained satellite systems at the current level of quantum computing technology.

Benchmarks: To evaluate the effectiveness of the lightweight QRL-SR by reducing the number of parameters, conventional RL-based LEO satellite routing algorithms that utilize classical NN serve as the benchmarks. This article aims to achieve lightweight implementation and enhanced efficiency through quantum techniques within the RL framework, by comparing the QRL-SR algorithm with conventional artificial intelligence (AI) methodologies, especially the RL-based LEO satellite routing methodologies. This article employs conventional RL-based LEO satellite routing algorithms as benchmarks to analyze and clearly demonstrate the advantages achieved by incorporating QNN within the RL framework. To encompass the full spectrum of RL algorithms, this article employs the most representative and recent RL methodologies as benchmarks. This article employs three major categories of RL algorithms as benchmarks: 1) value-based RL algorithm; 2) policy gradient-based RL algorithm; and 3) hybrid RL algorithm. These are detailed as follows.

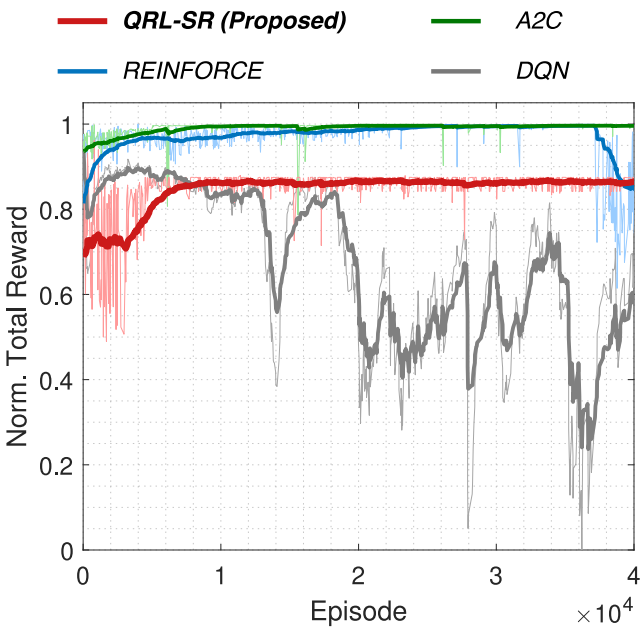

Fig. 5. Normalized total reward trends of the LEO satellites trained with the proposed algorithm and benchmarks over training episodes.

1) *Advantage Actor–Critic (A2C) [58]:* A2C is a hybrid RL algorithm that integrates value-based and policy gradient-based methods to optimize decision-making. Its actor component directly learns the policy while the critic evaluates the value function, with the advantage factor quantifying the relative merit of each action compared to a baseline and thereby reducing variance in policy updates. This combined approach leverages the strengths of both methodologies, enhancing stability and convergence in learning complex tasks. Consequently, A2C serves as a rigorous baseline for comparative studies against QRL-based approaches in LEO satellite routing.
2) *Reinforce [59]:* Reinforce is a policy gradient-based RL algorithm that directly updates policy parameters by maximizing the expected return. It employs classical NN architectures and relies on Monte Carlo sampling to provide unbiased estimates of the gradient, thereby facilitating stable policy improvements. Unlike QRL methods, Reinforce operates entirely within the classical computational paradigm. Consequently, it serves as a robust baseline for comparative studies in QRL-based LEO satellite routing.
3) *Deep Q-Network (DQN) [60]:* DQN is a value-based RL method that utilizes classical deep NN to approximate the optimal action-value function. It represents a significant advancement in conventional RL by enabling agents to learn directly from high-dimensional sensory

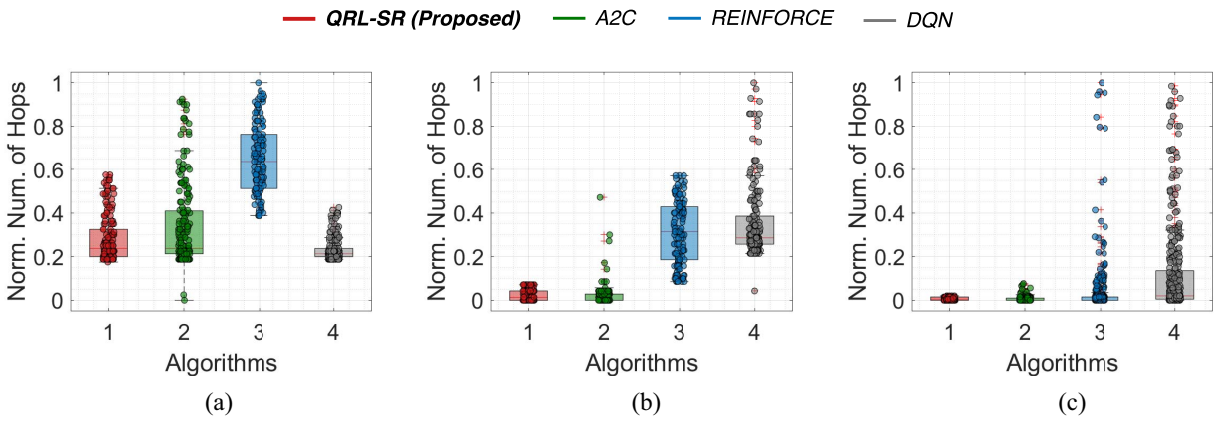

Fig. 6. Number of hops of the LEO satellites trained with the proposed algorithm and benchmarks. (a) Initial phase. (b) Intermediate phase. (c) Final phase.

inputs. Unlike QRL techniques, DQN employs traditional NN architectures within a classical computational framework. Its architecture incorporates mechanisms, such as experience replay and target networks to improve learning stability and convergence. Consequently, DQN serves as the robust baseline for comparative evaluations, including studies against QRL-SR in LEO satellite routing.

B. Result

1) Reward: Fig. 5 shows the trend of the normalized total rewards as the episode progresses. According to Fig. 5, the A2C has the highest reward value by achieving the highest reward of almost 1.0. After that, the reward value of QRL-SR is as high as 0.85, followed by the reward value of Reinforce. In A2C, QRL-SR, and Reinforce, the reward values all converge to the specific reward values. On the other hand, the DQN-based LEO satellite routing algorithm shows unstable training performance with fluctuations ranging from 0 to 0.8. In DQN, LEO satellites can not train the correct routing policy because the reward value does not converge to a specific value and continues to fluctuate.

2) Number of Hops: Fig. 6 shows the normalized routing hop count as a box plot according to the initial [Fig. 6(a)]/intermediate [Fig. 6(b)]/final episode phase [Fig. 6(c)]. In the initial phase, LEO satellites of the QRL-SR route to the least number of hops, similar to LEO satellites of the A2C. Although their average hop counts are similar, the hop count variance of the QRL-SR's LEO satellites is lower than that of the LEO satellites of A2C. That is, many QRL-SR's LEO satellites route with fewer hops than A2C's LEO satellites. Even in the intermediate phase, the LEO satellites of the QRL-SR and A2C still route to the lowest hop count. Unlike QRL-SR and A2C, which update training at specific time steps before the episode ends, Reinforce has a relatively large number of routing hops because it updates training only after the episode ends. Even in the final phase, the LEO satellites of the QRL-SR and AC succeed in routing with the least hops. The LEO satellites of the DQN fail to train and have the largest number of routing hops on average. In addition, DQN also has the largest variance in the number of routing hops, along with Reinforce. The small hop count simplifies routing tables and protocols, making managing and maintaining LEO satellite networks easier. It also reduces overall power consumption, improving the LEO satellite's lifespan and operational efficiency. From this point of view, the LEO satellites of the QRL-SR outperform the RL algorithm using the classical NN in terms of the number of hops required for routing.

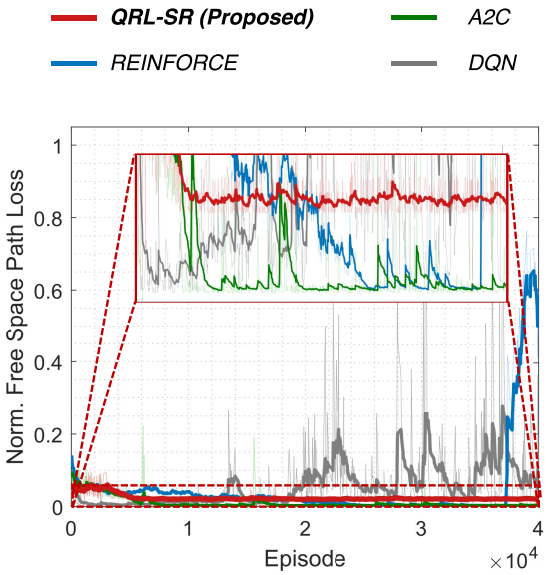

Fig. 7. Free space path loss of the LEO satellites trained with the proposed algorithm and benchmarks over training episodes.

3) Free Space Path Loss: Fig. 7 shows the normalized free space path loss trend as the episode progresses. It can be seen that the free space path loss of all routing algorithms except Reinforce and DQN converges stably. The LEO satellites of the Reinforce and DQN fail to train and have high free space path loss. The LEO satellites of the A2C have the smallest level of free space path loss. The free space path loss of the QRL-SR's LEO satellites is slightly higher than A2C, but they maintain a small level of the free space path loss and allow receiving LEO satellites to receive signals more strongly. The free space path loss refers to a decrease in signal strength

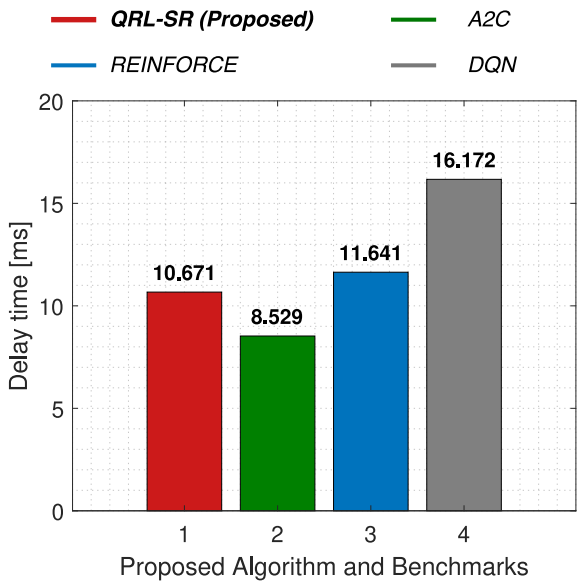

Fig. 8. Average delay time of the LEO satellites trained with the proposed algorithm and benchmarks.

TABLE III
NUMBER OF PARAMETERS ACCORDING TO THE PROPOSED ALGORITHM AND BENCHMARKS

	Norm. Parameters	# Num. Parameters
QRL-SR (Proposed)	**0.163%**	**56**
A2C	100%	34309
REINFORCE	98.872%	33922
DQN	98.872%	33922

during signal propagation, and a lower path loss indicates that a more robust signal can be received on the receiving side. Lowering the free-space path loss improves signal strength and quality in intersatellite communication, enabling efficient and reliable data transfer. This is an essential factor in optimizing the performance of the LEO satellite networks. In terms of free space path loss, it can be seen that the performance of the proposed QRL-SR algorithm is similar to or superior to that of other conventional RL.

4) Delay Time: Fig. 8 shows the average routing delay time of the proposed algorithm and benchmarks. The LEO satellites of the A2C have the smallest delay time of 8.529 ms. Next, the LEO satellites of the QRL-SR and Reinforce have delay times of 10.671 ms and 11.641 ms, respectively. The LEO satellites of the DQN have the largest delay of 16.172 ms. Because the delay time is accumulated when several hops are passed between LEO satellites, the overall delay time is reduced only when the delay at each hop is minimized. From this perspective, the LEO satellites of the QRL-SR are comparable to or outperforming the delay performance of RL algorithms using classical NN.

5) Number of Parameters: Table III shows the number of required training parameters and their normalized values of the proposed algorithm and benchmarks. The A2C, Reinforce, and DQN, which are algorithms using classical NN to train LEO satellites, all require over 30 000 parameters. Reinforce and DQN require 33 922 training parameters and even A2C requires 34 309 training parameters. On the other hand, the proposed QRL-SR requires only 56 parameters to train LEO satellites. QRL-SR reduces the training parameters by nearly $1000x$ compared to conventional RL-based LEO routing algorithms. Because QRL-SR uses QNN, it does not require large parameters like other LEO satellite routing algorithms that use classical NN.

In QRL-SR, each channel is represented by a single qubit, contributing to the efficiency of the QNN design. Specifically, each channel utilizes three rotation parameters (for rotations around the *RX*, *RY*, and *RZ* axes) to represent state information. This design spans two layers, enhancing the network's expressive power. The rotation parameters transform the qubit state around specific axes, enabling an accurate representation of complex quantum states. A bias parameter is also included for output fine-tuning, resulting in a toal of $3 \times 2 + 1 = 7$ parameters per channel. The parameter property plays a crucial role in enhancing computing and energy efficiency for LEO satellites. These parameters are crucial for optimizing computing and energy efficiency in LEO satellites. By adjusting resource allocation, routing protocols, and communication schedules, LEO satellites can reduce power consumption and improve computational efficiency. This approach demonstrates the potential of quantum computing to learn and represent complex state spaces with minimal resource use. The power of QRL-SR is to be able to exponentially reduce the number of traning parameters required to train LEO satellites with routing policies. This article succeeds in exponentially reducing the number of training parameters while showing routing performance similar to that of conventional RL-based LEO satellite routing. Reducing the number of parameters is an essential factor for LEO satellites as it allows computational resource-limited LEO satellite systems to be lightweight.

6) Routing Trajectories of the LEO Satellites: Fig. 9 shows the trajectories of the LEO satellites of the proposed algorithm and benchmarks as they route the data packets from the South Pole to the North Pole. All LEO satellites must route data packets from the South Pole to the North Pole without routing the malfunctioning LEO satellites. Fig. 9 shows the number of routing hops, such as P1, P2, . . ., PN. The color bar on the right side of each figure represents the altitude of the LEO satellites. It can be seen that QRL-SR, A2C, Reinforce, and DQN-based LEO satellites route to 8, 7, 15, and 20 hops, respectively. Here, the LEO satellites of the DQN fail to route to the North Pole. The relatively small delay time of the DQN in Fig. 8 is meaningless because it continues to route to nearby LEO satellites regardless of the direction from the source LEO satellite to the target LEO satellite. It can be seen that the LEO satellites of the QRL-SR efficiently route data packets from the South Pole to the North Pole with a small number of routing hops to avoid routing the malfunctioning LEO satellites. Even when checking the actual data packet routing trajectories, it can be intuitively confirmed that the routing of the QRL-SR's LEO satellites is more efficient than the routing of the conventional RL algorithm-based LEO satellites.

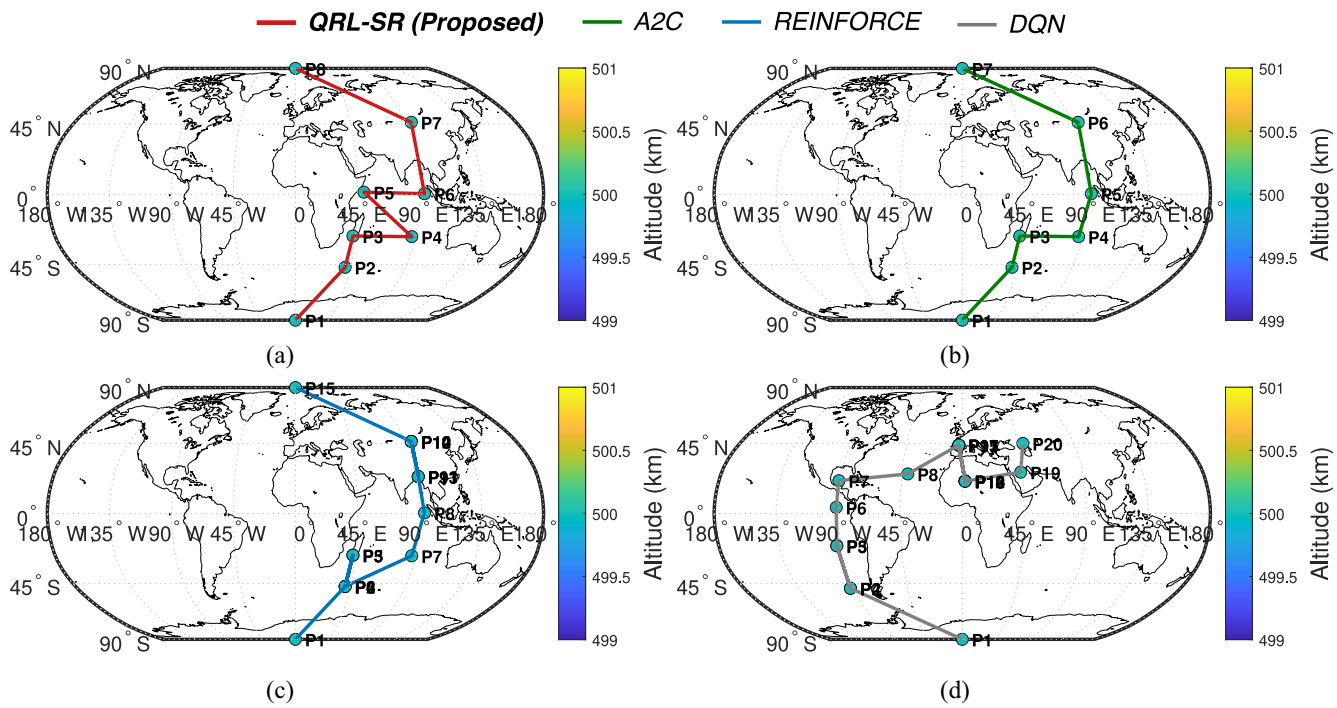

Fig. 9. Trajectories of the LEO satellites when routing from the South Pole to the North Pole. (a) QRL-SR. (b) A2C. (c) Reinforce. (d) DQN.

While the primary focus of this article is on enhancing routing efficiency and reducing the number of training parameters, the proposed QRL-SR algorithm also holds considerable potential for the broader landscape of LEO satellite network applications. Its integration with existing GEO and MEO satellite communication systems can facilitate a more cohesive and efficient global network architecture. The integration of existing GEO and MEO satellite systems with LEO networks enhances global network efficiency by leveraging their complementary capabilities. GEO satellites provide broad, stable coverage despite higher latency, while LEO satellites offer low latency and agile routing, and MEO systems serve as a bridge between these extremes. This multilayered architecture enables optimized routing, improved load balancing, and increased fault tolerance, resulting in a resilient and high-performance global network suitable for diverse operational conditions. This is particularly significant for providing high-speed Internet access to underserved regions and ensuring robust communication links during disaster response scenarios. By optimizing routing paths and reducing latency through intelligent decision-making, QRL-SR can contribute to the overall reliability and cost-efficiency of integrated satellite networks.

VII. Conclusion and Future Work

This research addresses the challenge of global Internet connectivity using LEO satellite networks and introduces the QRL-SR algorithm. Addressing the challenges of dynamic topology changes, frequent handovers, and strict SWaP constraints, the QRL-SR algorithm takes advantage of QNN to exponentially reduce training parameters. This efficiency enables real-time decision-making without overloading the limited on-board computational resources of the LEO satellites. Simulations using actual LEO satellite data, i.e., TLE, demonstrate that the QRL-SR algorithm matches or outperforms the routing performance of the conventional RL methods while significantly reducing computational requirements. This quantum approach offers a scalable and efficient solution for the dynamic topology of LEO constellations, advancing the development of resilient NTN. This quantum approach offers an efficient and lightweight routing solution for the dynamic topology of the LEO satellite constellation networks, contributing to the realization of seamless global Internet connectivity.

As future work, it is worth considering the application of QRL-SR in mega-constellations that employ thousands of LEO satellites. By integrating QNNs that leverage quantum superposition, entanglement, and parallelism, QRL-SR facilitates scalable management of the high-dimensional state spaces inherent in mega-constellation LEO satellite routing, thereby offering significant advantages over conventional RL approaches.

References

[1] S. Park, G. S. Kim, Z. Han, and J. Kim, "Quantum multi-agent reinforcement learning is all you need: Coordinated global access in integrated TN/NTN cube-satellite networks," *IEEE Commun. Mag.*, vol. 62, no. 10, pp. 86–92, Oct. 2024.

[2] Z. Gao, A. Liu, C. Han, and X. Liang, "Max completion time optimization for Internet of Things in LEO satellite-terrestrial integrated networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9981–9994, Jun. 2021.

[3] J. Chu, X. Chen, C. Zhong, and Z. Zhang, "Robust design for NOMA-based multibeam LEO satellite Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1959–1970, Feb. 2021.

[4] A. K. Dwivedi, S. Chaudhari, N. Varshney, and P. K. Varshney, "Performance analysis of LEO satellite-based IoT networks in the presence of interference," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8783–8799, Mar. 2024.

[5] M. Hu, M. Xiao, W. Xu, T. Deng, Y. Dong, and K. Peng, "Traffic engineering for software-defined LEO constellations," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 5090–5103, Dec. 2022.

[6] Z. Zhang et al., "User activity detection and channel estimation for grant-free random access in LEO satellite-enabled Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8811–8825, Sep. 2020.

[7] F. Tang, H. Zhang, and L. T. Yang, "Multipath cooperative routing with efficient acknowledgement for LEO satellite networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 179–192, Jan. 2019.

[8] J. Zhou, Q. Yang, L. Zhao, H. Dai, and F. Xiao, "Mobility-aware computation offloading in satellite edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9135–9149, Oct. 2024.

[9] Z. Han, C. Xu, G. Zhao, S. Wang, K. Cheng, and S. Yu, "Time-varying topology model for dynamic routing in LEO satellite constellation networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3440–3454, Mar. 2023.

[10] X. Zhang et al., "Cost-effective hybrid computation offloading in satellite-terrestrial integrated networks," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 36786–36800, Nov. 2024.

[11] S. Park, G. Seon Kim, S. Jung, and J. Kim, "Markov decision policies for distributed angular routing in LEO mobile satellite constellation networks," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38744–38754, Dec. 2024.

[12] J. Cao, S. Zhang, Q. Chen, H. Wang, M. Wang, and N. Liu, "Computing-aware routing for LEO satellite networks: A transmission and computation integration approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16607–16623, Dec. 2023.

[13] S. S. Hassan, Y. M. Park, Y. K. Tun, W. Saad, Z. Han, and C. S. Hong, "Satellite-based ITS data offloading & computation in 6G networks: A cooperative multi-agent proximal policy optimization DRL with attention approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4956–4974, May 2024.

[14] G. S. Kim, J. Chung, and S. Park, "Realizing stabilized landing for computation-limited reusable rockets: A quantum reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 12252–12257, Aug. 2024.

[15] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, "Variational quantum circuits for deep reinforcement learning," *IEEE Access*, vol. 8, pp. 141007–141024, 2020.

[16] S. Park, J. P. Kim, C. Park, S. Jung, and J. Kim, "Quantum multi-agent reinforcement learning for autonomous mobility cooperation," *IEEE Commun. Mag.*, vol. 62, no. 6, pp. 106–112, Jun. 2024.

[17] Y. Huang, X. Jiang, S. Chen, F. Yang, and J. Yang, "Pheromone incentivized intelligent multipath traffic scheduling approach for LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 5889–5902, Aug. 2022.

[18] D. Bhattacharjee et al., "On-demand routing in LEO mega-constellations with dynamic laser inter-satellite links," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 5, pp. 7089–7105, Oct. 2024.

[19] C. Li, W. He, H. Yao, T. Mai, J. Wang, and S. Guo, "Knowledge graph aided network representation and routing algorithm for LEO satellite networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5195–5207, Apr. 2023.

[20] E. Ekici, I. Akyildiz, and M. Bender, "A multicast routing algorithm for LEO satellite IP networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 183–192, Apr. 2002.

[21] Y. Huang, B. Feng, A. Tian, P. Dong, S. Yu, and H. Zhang, "An efficient differentiated routing scheme for MEO/LEO-based multi-layer satellite networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 1026–1041, Jan./Feb. 2024.

[22] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen, "Analysis of inter-satellite link paths for LEO mega-constellation networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2743–2755, Mar. 2021.

[23] J. W. Rabjerg, I. Leyva-Mayorga, B. Soret, and P. Popovski, "Exploiting topology awareness for routing in LEO satellite constellations," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[24] X. Feng, Y. Sun, and M. Peng, "Distributed satellite-terrestrial cooperative routing strategy based on minimum hop-count analysis in mega LEO satellite constellation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 11, pp. 10678–10693, Nov. 2024.

[25] F. Yan, Z. Wang, S. Zhang, Q. Meng, and H. Luo, "Logic path identified hierarchical routing for large-scale LEO satellite networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 4, pp. 3731–3746, Jul./Aug. 2024.

[26] L. Zhang, Y. Du, Y. Zhang, and Y. Tang, "Distributed anti-cascading routing scheme based on fuzzy logic in LEO satellite networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 2, pp. 3196–3211, Feb. 2025.

[27] P. Zuo, C. Wang, Z. Yao, S. Hou, and H. Jiang, "An intelligent routing algorithm for LEO satellites based on deep reinforcement learning," in *Proc. Veh. Technol. Conf.*, Norman, OK, USA, Sep. 2021, pp. 1–5.

[28] X. Liu, H. Zhou, Z. Zhang, Q. Gao, and T. Ma, "Multipath cooperative routing in ultradense LEO satellite networks: A deep-reinforcement-learning-based approach," *IEEE Internet Things J.*, vol. 12, no. 2, pp. 1789–1804, Jan. 2025.

[29] H. Liu, J. Lai, J. Zhu, L. Gan, and Z. Chang, "Enabling high-throughput routing for LEO satellite broadband networks: A flow-centric deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 11, no. 17, pp. 28705–28720, Sep. 2024.

[30] C. Park et al., "Quantum multiagent actor–critic networks for cooperative mobile access in multi-UAV systems," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 20033–20048, Nov. 2023.

[31] W. J. Yun, J. P. Kim, S. Jung, J.-H. Kim, and J. Kim, "Quantum multiagent actor–critic neural networks for Internet-connected multi-robot coordination in smart factory management," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9942–9952, Jun. 2023.

[32] Z. Yang, H. Li, Q. Wu, and J. Wu, "Analyzing and optimizing BGP stability in future space-based Internet," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, San Diego, CA, USA, Dec. 2017, pp. 1–8.

[33] P. Truchly and M. Urbanovic, "MPLS throughput over GEO satellites," in *Proc. IEEE Int. Symp. Electron. Mar. (ELMAR)*, Zadar, Croatia, Jun. 2006, pp. 305–308.

[34] Y. Jian, W. Shining, and Z. Feng, "Label Mis-switching rate analysis in satellite MPLS networks," in *Proc. IEEE Int. Conf. Wireless Commun. Signal Process.*, Nanjing, China, Nov. 2009, pp. 1–4.

[35] A. Markhasin, "Ubiquitous and multifunctional mobile satellite all-IP over DVB-S networking technology 4G with radically distributed architecture for RRD regions," in *Proc. IEEE Int. Workshop Satell. Space Commun.*, Salzburg, Austria, Sep. 2007, pp. 99–103.

[36] Y. Wu, Z. Yang, and Q. Zhang, "A novel DTN routing algorithm in the GEO-relaying satellite network," in *Proc. IEEE Int. Conf. Mobile Ad-hoc Sensor Netw. (MSN)*, Shenzhen, China, Dec. 2015, pp. 264–269.

[37] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay- and disruption-tolerant networking (DTN): An alternative solution for future satellite networking applications," *Proc. IEEE*, vol. 99, no. 11, pp. 1980–1997, Nov. 2011.

[38] R. C. Kizilirmak, I. E. Ehile, B. Kabdrashev, and S. Khvan, "Enhancing inter-satellite data relay in dynamic space communication," in *Proc. IEEE Int. Conf. Adv. Commun. Technol. (ICACT)*, Pyeong Chang, South Korea, Feb. 2024, pp. 1–5.

[39] H. Yang et al., "Interruption tolerance strategy for LEO constellation with optical inter-satellite link," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 4, pp. 4815–4830, Dec. 2023.

[40] Y. Kwak, W. J. Yun, S. Jung, and J. Kim, "Quantum neural networks: Concepts, applications, and challenges," in *Proc. Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jeju Island, South Korea, Aug. 2021, pp. 413–416.

[41] H. Baek, S. Park, and J. Kim, "Logarithmic dimension reduction for quantum neural networks," in *Proc. Int. Conf. Inf. Knowl. Manag. (CIKM)*, New York, NY, USA, 2023, pp. 3738–3742.

[42] F. Schmidt-Kaler et al., "Realization of the Cirac–Zoller controlled-NOT quantum gate," *Nature*, vol. 422, no. 6930, pp. 408–411, Mar. 2003.

[43] "Celestrack." Accessed: Oct. 20, 2024. [Online]. Available: https://celestrak.org/

[44] "Space-track." Accessed: Oct. 20, 2024. [Online]. Available: https://www.space-track.org/auth/login

[45] F. S. Prol et al., "Position, navigation, and timing (PNT) through low earth orbit (LEO) satellites: A survey on current status, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 83971–84002, 2022.

[46] R. Bhusal and K. Subbarao, "Generalized polynomial chaos-based ensemble Kalman filtering for orbit estimation," in *Proc. Amer. Control Conf. (ACC)*, New Orleans, LA, USA, May 2021, pp. 4290–4295.

[47] *Admiralty Manual of Navigation*, HM Stationery Office, London, U.K., 1987.

[48] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 38, no. 5, pp. 1207–1220, Oct. 2008.

[49] S. Park et al., "Joint quantum reinforcement learning and stabilized control for spatio-temporal coordination in metaverse," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12410–12427, Dec. 2024.
[50] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer, "Decentralized control of partially observable Markov decision processes," in *Proc. IEEE Conf. Decision Control (CDC)*, Firenze, Italy, Dec. 2013, pp. 2398–2405.
[51] G. S. Kim, S. Lee, T. Woo, and S. Park, "Cooperative reinforcement learning for military drones over large-scale battlefields," *IEEE Trans. Intell. Veh.*, early access, Oct. 9, 2024, doi: 10.1109/TIV.2024.3472213.
[52] 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.*, early access, Mar. 28, 2025, doi: 10.1109/TAES.2025.3556050.
[53] M. Hu, M. Xiao, Y. Hu, C. Cai, T. Deng, and K. Peng, "Software defined multicast using segment routing in LEO satellite networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 835–849, Jan. 2024.
[54] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Denver, CO, USA, Nov. 1999, pp. 1008–1014.
[55] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, "General parameter-shift rules for quantum gradients," *Quantum*, vol. 6, pp. 677–703, Mar. 2022.
[56] G. S. Kim, S. Y.-C. Chen, S. Park, and J. Kim, "Quantum reinforcement learning for coordinated satellite systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2025, pp. 1–5.
[57] N. W. Hendrickx et al., "A four-qubit germanium quantum processor," *Nature*, vol. 591, no. 7851, pp. 580–585, Mar. 2021.
[58] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 48, New York, USA, Jun. 2016, pp. 1928–1937.
[59] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Lang.*, vol. 8, pp. 229–256, May 1992.
[60] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

Gyu Seon Kim received the B.S. degree in aerospace engineering from Inha University, Incheon, Republic of Korea, in 2023. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea.

His research focuses include deep reinforcement learning algorithms and their applications to autonomous mobility systems.

Mr. Kim received the IEEE Seoul Section Student Paper Contest Award (2023).

Sungjoon Lee received the B.S. degree in electronics engineering from Soongsil University, Seoul, Republic of Korea, in 2024. He is currently pursuing the M.S. degree with the Department of Electrical and Computer Engineering, Korea University, Seoul.

His research focuses include deep reinforcement learning algorithms and their applications to autonomous mobility systems.

In-Sop Cho received the B.S. and Ph.D. degrees in radio communications engineering from Korea University, Seoul, Republic of Korea, in 2011 and 2021, respectively.

In 2021, he was with Korea Electric Power Research Institute, Daejeon, Republic of Korea, as a Senior Researcher. Since 2022, he has been with Electronics and Telecommunications Research Institute (ETRI), Daejeon, where he is currently a Senior Researcher. His research interests include resource optimization for wireless communication, network scheduling, and satellite communication.

Soohyun Park (Member, IEEE) received the B.S. degree in computer science and engineering from Chung-Ang University, Seoul, Republic of Korea, in February 2019, and the Ph.D. degree in electrical and computer engineering from Korea University, Seoul, in August 2023.

She has been an Assistant Professor with Sookmyung Women's University, Seoul, since March 2024. She was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Korea University from September 2023 to February 2024.

Dr. Park was a recipient of the ICT Express Best Reviewer Award in 2021, the IEEE Seoul Section Student Paper Contest Awards, and the IEEE Vehicular Technology Society (VTS) Seoul Chapter Awards.

Joongheon Kim (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and engineering from Korea University, Seoul, Republic of Korea, in 2004 and 2006, respectively, and the Ph.D. degree in computer science from the University of Southern California at Los Angeles, Los Angeles, CA, USA, in 2014.

He has been with Korea University since 2019, where he is currently an Associate Professor with the School of Electrical Engineering and an Adjunct Professor with the Department of Communications Engineering and the Department of Semiconductor Engineering. He has also been the Director for the Net-Zero CAFE (Connectivity and Autonomy for Future Ecosystem) Research Center, Seoul, sponsored by the Korean Ministry of Science and ICT (MSIT), since 2024. Before joining Korea University, he was a Research Engineer with LG Electronics, Seoul, from 2006 to 2009, a Systems Engineer with Intel Corporation, Santa Clara, CA, USA, from 2013 to 2016, and an Assistant Professor with Chung-Ang University, Seoul, Republic of Korea, from 2016 to 2019.

Dr. Kim was a recipient of the Annenberg Graduate Fellowship with the Ph.D. admission from USC in 2009, the Intel Corporation Next Generation and Standards (NGS) Division Recognition Award in 2015, the IEEE Systems Journal Best Paper Award in 2020, the IEEE ComSoc Multimedia Communications Technical Committee in MMTC Outstanding Young Researcher Award in 2020, the IEEE ComSoc MMTC Best Journal Paper Award in 2021, and the Granite Tower Best Research Award for Top 3% Research and Development Achievement at Korea University in 2024. He also received several awards from IEEE conferences, including the IEEE ICOIN Best Paper Award in 2021 and the IEEE ICTC Best Paper Award in 2022. He serves as an Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE INTERNET OF THINGS JOURNAL.