IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2024, accepted 14 April 2024, date of publication 16 April 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3390042

RESEARCH ARTICLE

Quantum Reinforcement Learning for Spatio-Temporal Prioritization in Metaverse

SOOHYUN PARK[1], **HANKYUL BAEK**[2], **AND JOONGHEON KIM**[2], **(Senior Member, IEEE)**
[1]Division of Computer Science, Sookmyung Women's University, Seoul 04310, Republic of Korea
[2]Department of Electrical and Computer Engineering, Korea University, Seoul 02841, Republic of Korea

Corresponding authors: Hankyul Baek (67back@korea.ac.kr) and Joongheon Kim (joongheon@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea under Grant 2022R1A2C2004869, and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government [Ministry of Science and Information and Communications Technology (MSIT)] (Intelligent 6G Wireless Access System) under Grant 2021-0-00467.

ABSTRACT A metaverse is composed of a physical-space and virtual-space, with the aim of having users in both the virtual reality and the real world experience. Prioritization is essential, but it is not straight-forwarded due to the limitation of computing resources in real-world, making it impossible to synchronize all data. Therefore, it is crucial to allocate resources based on regional preferences in physical-space and content preferences of users in virtual-space. The referencing system consists of two-stage sub-computations, 1) spatial-prioritization for more data gathering under the consideration of avatar-popularity one top of physical-space and 2) temporal-prioritization for virtual-space rendering under the avatar-popularity. Both prioritization tasks are combinatorics problems and are well-known NP-hard. The problem scale is also large, making it difficult to solve within given times. The classical deep learning cannot be the solution. Quantum-based learning algorithms can be the potential solutions due to high-performance computing capabilities, because a small number of qubits can represent an exponentially large amount of information. On top of these advantages, an improved quantum reinforcement learning (QRL) algorithm is proposed for reducing the control dimensions into a logarithmic scale. We corroborate that our proposed QRL-based algorithm for low-dimensional spatio-temporal prioritization improves convergence and performance.

INDEX TERMS Metaverse, scheduling, quantum neural network, quantum reinforcement learning.

I. INTRODUCTION

Recently, the metaverse has been establishing itself as a contact-free social interaction service. The metaverse is a space where the real world and virtual space coexist, allowing users to experience both real and virtual environments. It has been utilized in various applications, ranging from company offices and classrooms to museums and concert halls [1]. In order to improve service quality, metaverse platforms have been extensively researched. Starting with early virtual space such as GatherTown, which provided 2D pixel art-style virtual environments [2], the development has progressed to metaverse platforms built on actual 3D maps [3], [4] and further to digital twins achieved through synchronization between the real world and virtual reality using multiple sensors [5]. This evolution has been aimed at enhancing the user's sense of realism. As the metaverse provides services that allow users to experience 3D media experiences, the amount of data exchanged increases significantly [6]. In addition, to enhance the user's sense of realism, low-latency communication such as the tactile Internet must be ensured. Due to the presence of these technical hurdles, ongoing discussions are being held in both industry and academia.

One of main differences between the metaverse system and the other systems is that the metaverse's places where users physically exist have no association with the places where they are interested in [7]. In conventional social media

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

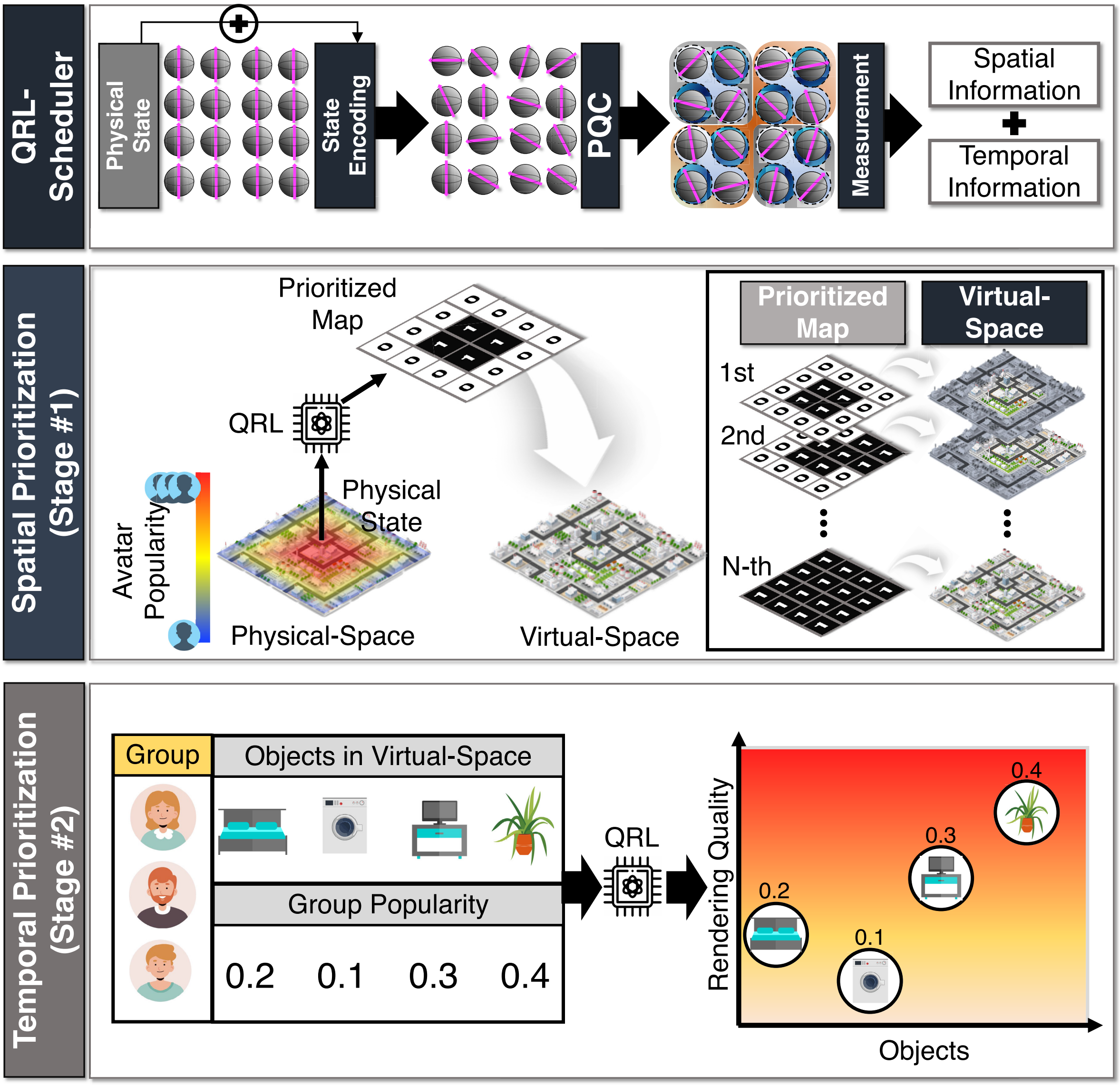


FIGURE 1. Overview of QRL scheduling for spatio-temporal prioritization.

platforms, the places where the users exist are the ones where they are also interested in. For example, the users on social networks tag their locations because they are physically present and interested. However, in metaverse-based concerts, the users are generally at home while their associated avatars are located at the concert halls. In this paper, the metaverse focuses on a 3D immersive experience involving various physical space. As the physical space changes over time due to sensor information, synchronization between the physical and virtual space is crucial. However, as previously mentioned, transmitting 3D data requires tremendous network throughput, making it impossible to stream 3D data to entire areas. Consequently, this paper proposes two consecutive methods: (i) data gathering scheduling based on avatar-popularity in physical space and (ii) improving user service quality within the virtual space. As depicted in Fig. 1, users in the real world request services in various virtual space. By collecting these requests, the central server calculates the avatar's demand, where the avatar is the user identity in virtual space. The statistical distribution of user interests is referred to as *avatar-popularity*. Then, the central server collects whether to receive the physical-space's data. Improving the avatar-popularity and reducing the gap between physical and virtual space is achieved through a process called *spatial-prioritization*. Furthermore, when users experience services within virtual reality, users with similar viewpoints are grouped, creating a *group-popularity* for specific content. The sampling ratio for viewpoints is then adjusted to provide the service, while non-prioritized areas are not updated. This corresponds to the *temporal-prioritization*. By implementing these two processes, the metaspace can achieve high *spatio-temporal prioritization*.

However, these problems are combinatorial in nature and are well-known to be NP-hard [8]. As a result, they cannot be solved within a given time as the problem size increases. To address this, standard deep learning can be employed to find sub-optimal solutions. Although standard deep learning has demonstrated effectiveness in combinatorial problems, the model convergence and performance deteriorate as the control size increases [9]. To tackle this issue, we explore the potential of quantum computing [10]. Recent advancements in quantum computing has theoretically enabled the quadratic acceleration of combinatorial problems [11]. Nevertheless, scalability concerning problem size remains a challenge. When a large-scale quantum computer attempts to solve optimization problems, quantum noise impedes the attainment of suboptimal solutions. Consequently, the quantum computing system is a noisy intermediate-scale quantum (NISQ) device [12].

In this paper, we revisit quantum neural network (QNN) based quantum reinforcement learning (QRL) and significantly enhance the scalability of control sizes in spatio-temporal prioritization [13], [14], [15], [16]. Baek et al. employed a specific measurement method called *basis measurement (BM)* [17]. By leveraging BM, a small number of quantum bits (qubits) can represent a exponentially large amount of information compared to classical bits, allowing for more efficient computation and data storage. Motivated by this, the resources can be managed with logarithmic output size. Motivated by this, we re-design a BM-suitable quantum actor critic for QNN-based QRL to logarithmically reduce the control size. On top of the proposed QRL algorithm, both spatio-temporal prioritization can be realized even if the control size increases. This approach can be a milestone in order to realize the advantages of QNN on top of metaverse systems, which can be one of emerging future research directions.

A. CONTRIBUTIONS

The major contributions of this study can be summarized as follows.

- First of all, this research is the first attempt to characterize the space separation between physical-space and virtual-space in metaverse systems. Based on this key characteristic in metaverse, the corresponding objective aims at the virtual-space quality maximization subject to physical-space constraints.
- In addition, the proposed algorithm is the inaugural approach to employ low-complexity quantum actor-critic neural networks for scalable *spatial* prioritization in physical-space data gathering. As well-studied in [17], the use of quantum neural network algorithms can realize output dimension reduction into a logarithmic-scale which introduce low-complexity computation.
- Moreover, the algorithm takes into account avatar-popularity for differentiated quality control in constructing high-reality virtual-space under limited physical-space resources concerning *temporal* prioritization.
- Finally, an innovative system-wide approach is introduced for faster *spatio-temporal* prioritization to fully utilize the realistic virtual worlds, in real-time.

B. ORGANIZATION

The rest of this paper is organized as follows. Sec. II illustrates the basic concepts of metaverse network architecture and related work. Sec. III presents our proposed QRL-based spatio-temporal prioritization in metaverse systems. Sec. IV evaluates the performance of our proposed algorithm and Sec. V concludes this paper.

II. PRELIMINARIES

A. METAVERSE NETWORK ARCHITECTURE

Metaverse system consists of two different spaces, named physical-space and virtual-space [18]. The physical-space is a target space that the metaverse aims to mimic. The virtual-space is a result space where the users want to interact or see the objects which are physically apart. Due to the advantage of a virtual-space that can reflect anywhere and anytime, users can break the spatio-temporal limitations. However, there is a huge challenge to service the metaverse efficiently as the realistic metaverse utilizes 3D contents format, such as points cloud and depth images, due to their realistic representation ability [19]. However, there is still a challenge in processing these extensive data with small delay [20]. These delays primarily occur in the two-stage prioritization process that precedes the provision of metaverse services.

- **Spatial-Prioritization.** When updating the virtual-space to reflect the physical-space, the most naïve approach is to update all target physical-spaces for all users. However, this method is not suitable for serving the metaverse in a resource-constrained environment [21]. Consequently, spatial-prioritization optimizes metaverse establishment by synchronizing in a manner that considers the popularity of avatars, the virtual-space representations of users [22]. By allocating more computing and communication resources to update virtual hotspots with high avatar popularity, the user experience of a larger number of users can be enhanced [23].
- **Temporal-Prioritization.** Temporal-prioritization is a method of prioritizing physical-space and virtual-space by considering group popularity. Even if avatars are in the same spatial area, the objects in the virtual-space with which they interact in reality may differ over time. Accordingly, this approach groups users with similar preferences at each time point and adjusts the quality of objects within the virtual-space by reflecting the popularity of the respective group. This method, like frustum culling commonly used in resource-intensive games, selectively provides only the necessary objects within the virtual-space to the group, preventing unnecessary

use of computing and communication resources and contributing to the overall stability of the system [24].

Motivated by this, we design avatar popularity and group popularity in each prioritization process as referred to Sec. III-D2. We design the metaverse system based on these factors in this paper.

B. RELATED WORK

In recent years, AI-based system optimization can be categorized into two main fields: i) real-world system optimization and ii) metaverse system optimization. In real-world optimization, AI is widely used to minimize the required cost of the system. In real-world optimization, AI is widely used to minimize the required cost of the system. In [25], AI is leveraged to predict future load consumption based on historical data. Tanwar et al. [26], adopts AI techniques to maintain the robustness of the system by identifying energy theft, thereby preventing cost leakage. Kumari et al. [27] leverage AI to optimize the scheduling algorithm for charging and discharging in electrical vehicle environments. On the other hand, several studies leverage AI for metaverse system optimization. In [28], an extensive survey of metaverse technologies is conducted, addressing key challenges, security, and privacy. The fundamental architectures and components of the metaverse are presented in [29], where user interactions, implementations, and applications are also explored. Additionally, blockchain-based metaverse technologies are introduced and proposed in various research results, i.e., fusing of building information modeling and blockchain for metaverse [30], fusing blockchain and AI with metaverse [31], blockchain-aided secure semantic communications for AI-generated metaverse contents [32], and promoting the sustainability of blockchain in Web 3.0 and the metaverse via incentive mechanisms [33]. Furthermore, Meng et al. [34] underscore a critical technical challenge pertinent to this paper and propose a corresponding solution. The solution relies on prediction mechanisms, while the proposed algorithm offers a system-wide solution encompassing data-gathering and the quality of experience. Lastly, in [35], as an example of metaverse systems, financial crimes in Web 3.0-empowered metaverse is introduced in terms of taxonomy, countermeasures, and research opportunities.

C. QUANTUM REINFORCEMENT LEARNING

Fig. 2(a) illustrates the operations of a standard QNN. The primary objective of machine learning is to approximate target values based on given data points. In classical NNs, the encoding process is straightforward because both trainable parameters and data points are real values in Euclidean space. However, processing data points in QNNs is not straightforward, as data points exist in Euclidean space while quantum states reside in complex Hilbert space. To address this, the state encoder in QNNs utilizes rotation gates, contrasting with the linear operations on unbounded real values employed in classical NNs. In QNNs, data points are mapped to quantum states by transforming their positions using rotation gates, facilitating quantum computing. Indeed, quantum states are mapped onto the surface of a 3-dimensional unit sphere, as depicted in Fig. 2(b), and can be processed with rotation gates along the x, y, and z axes. Entanglement is a crucial aspect of quantum computing due to its inherent non-linearity. Entanglement establishes connections between distinct quantum systems and is analogous to the non-linear activation functions utilized in classical NNs. Furthermore, the universal function approximation theorem relies on non-linearity. Without quantum entanglement, measurements would be constrained to a linear structure, impeding the realization of quantum advantages. Consequently, entanglement assumes a significant role in quantum machine learning (QML), enabling quantum computers to perform intricate calculations and enhance machine learning algorithms.

The measurement process is indispensable for extracting information from quantum states since direct observation of these states is unfeasible. To obtain the outputs of QNNs, measurements must be conducted to convert probabilistic outcomes into deterministic ones [36]. Baek et al. have discovered that BM provides a more flexible measurement approach in quantum machine learning, as it enables the acquisition of probabilistic information about all potential quantum states [17]. As a result, probabilistic policies in quantum machine learning can achieve superior scalability compared to classical alternatives.

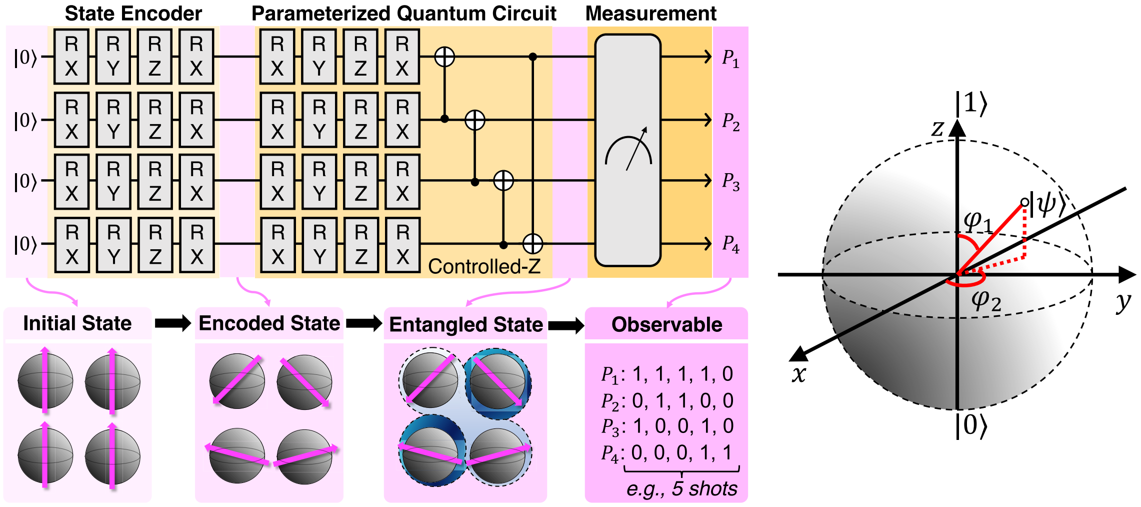
Given the aforementioned insights, we will adopt BM in QRL to improve the scalability and control size of probabilistic policies in contrast to classical methods.

III. QRL-BASED SPATIO-TEMPORAL PRIORITIZATION IN METAVERSE

A. OVERALL PROCEDURE AND OBJECTIVE

Our proposed algorithm consists of following two stages.

- *Spatial-Prioritization:* Due to the dynamic nature of physical-space information changing over time, an inevitable gap arises between virtual-space information and physical-space information. To address this issue, the concept of spatial prioritization is introduced (refer to Sec. III-D1), which enables the updating of physical-space information in prioritized regions with a high concentration of avatars (*i.e.*, high avatar popularity) to virtual-space information. By conducting real-time spatial prioritization, the objective is to reduce the spatial gap in age-of-information, focus on prioritization in areas with high avatar popularity, and simultaneously minimize the communication cost for updating the regions.
- *Temporal-Prioritization:* In temporal prioritization (refer to Sec. III-D2), the objective is to provide high-quality service to users for regions with temporal prioritization. It should be noted that delays occur due to not providing high-quality service for areas without temporal prioritization. To achieve this, when users receive metaverse services, they prefer certain items



(a) Quantum neural network.

(b) Bloch sphere.

FIGURE 2. Quantum neural networks and geometric qubit representation.

and backgrounds, and the user-preferred information is grouped to create group popularity. The popularity of a group represents the preference for each content, and adjusting the rendering ratio based on the preference, it plays a role in enhancing the quality. In this paper, the aim is to promote an improved sense of immersion quality based on the preferences of the entire user population for the virtual space being served while ensuring that the computing queue does not overflow.

B. OBJECTIVE

The main objective is about to maximize avatar-popularity, minimize communication costs, and minimize the gap between physical-space and virtual-space when gathering information from the physical-space. Moreover, it aims to maximize user immersion quality and minimize delay by adjusting the sampling rate to reflect group-popularity while rendering each user's interests.

C. QUANTUM ACTOR-CRITIC NETWORKS DESIGN

1) ALGORITHM CONCEPTS

Our proposed QRL algorithm is fundamentally based on actor-critic networks where the actor and critic networks are designed by QNNs. Our considering QNN-based QRL architecture consists of state encoding, parameterized quantum circuit (PQC), and lastly, measurement. For state encoding, the classical data/information will be encoded as the data/information over quantum domain. After the state encoding procedure, PQC will be executed which equivalent to the computation in hidden layers. Therefore, according to the fact that hidden layers are for linear and nonlinear transformation of given data for neural network-based function approximation [37], PQC is also for linear and nonlinear transformation over quantum domain by utilizing rotation gates and entanglement gates, respectively. Finally, measurement is for the deterministic computation for the result of PQC in order to derive final output, i.e., actions in RL computation. Here, the input and output data will be state and action data due to the nature of QRL/RL.

This QRL architecture is beneficial in terms of high-scalability because it can reduce RL action dimensions into a logarithmic-scale [14]. This is further beneficial in our considering metaverse application because it has huge action dimensions as massive users can exist over metaverse virtual-space.

2) ALGORITHM DETAILS

In our proposed algorithm, we assume that our proposed algorithm should work even if the number of available qubits is limited. This assumption is realistic because our proposed QNN-based actor-critic RL algorithm is fundamentally for NISQ-aware algorithm which can work with strictly limited number of qubits [38].

Within our metaverse system, the computational burden increases exponentially, corresponding to the expansion of virtual-space due to the curse of dimensionality [39]. Moreover, employing naïve QRL-based scheduling through QNN in metaverse systems is difficult to use in real-life, as the action dimension within QNN is larger than the number of qubits in NISQ era [40], [41].

Therefore, it is essential to design a new scheduling algorithm under the consideration of high-dimensional decision-making in a quantum domain. Thus, this paper revisits the parameterized quantum policy [42], and the QNN actor is defined as follows:

Definition 1 (QNN Actor): Given a QNN acting on Q qubits, taking as input state $s \in \mathbb{R}^{|s|}$, rotation angles $\boldsymbol{\theta} \in [0, 2\pi]^{|\boldsymbol{\theta}|}$, such that its corresponding unitary $U(s, \boldsymbol{\theta})$

produces the quantum state $|\psi_{s,\boldsymbol{\theta}}\rangle = U(s,\boldsymbol{\theta})|0^{\otimes Q}\rangle$, we define its associated QNN-based policy as:

$$\pi_{\boldsymbol{\theta}}(a|s) = \langle P_a \rangle_{s,\boldsymbol{\theta}} \tag{1}$$

where $\langle P_a \rangle_{s,\boldsymbol{\theta}}$ is the expectation value of a projection P_a associated to action a, such that $\sum_a P_a = I$ and $P_a P_{a'} = \mathbf{0}$. The variable $\boldsymbol{\theta}$ constitute all of its trainable parameters.

It should be noted that the number of projections P_a can be extended to the exponential of the number of qubits. In the proposed metaverse system, 2^Q actions are required as a consequence of the QRL-based spatial-prioritization and temporal-prioritization. It is evident that the number of actions is extremely large. In the QNN actor, the output size can be expanded to 2^Q. As a result, the number of qubits can be reduced logarithmically, which facilitates successful spatio-temporal prioritization. To efficiently train the quantum actor, we define the quantum critic as follows:

Definition 2 (QNN Critic): Given a QNN acting on multiple qubits, taking as input state $s \in \mathbb{R}^{|s|}$ and corresponding actions a, rotation angles $\boldsymbol{\phi} \in [0, 2\pi]^{|\boldsymbol{\phi}|}$, such that its corresponding unitary $U(s,\boldsymbol{\phi})$ approximates the action values, i.e., $V(s) = \langle P \rangle_{s,\boldsymbol{\phi}}$.

For more effective training of the QNN actor, the QNN critic evaluates the optimality of the prioritization determined by the QNN actor. The temporal difference (TD) is adopted as the objective and loss functions, which can be formulated as follows:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_s[\sum_{(s,a,r,s')\in\mathscr{B}} \delta_{\boldsymbol{\phi}}(s,a,s') \cdot \nabla_{\boldsymbol{\theta}} \log \pi(a|s;\boldsymbol{\theta})] \tag{2}$$

$$\nabla_{\boldsymbol{\phi}} \mathscr{L}(\boldsymbol{\phi}) = \sum_{(s,a,r,s')\in\mathscr{B}} \left\| \nabla_{\boldsymbol{\phi}} \delta_{\boldsymbol{\phi}}(s,a,s') \right\|^2 \tag{3}$$

such that

$$\delta_{\boldsymbol{\phi}}(s,a,s') = R(s,a) + \gamma V_{\boldsymbol{\phi}}(s') - V_{\boldsymbol{\phi}}(s), \tag{4}$$

where $R(s,a)$ denotes the reward function. Here, (2) represents an objective function for the QNN actor, while (3) denotes a loss function for the QNN critic. Here, the *Bellman optimality* in (4) is utilized for computation of the TD error. During the process of the loss function minimization with the TD error of the QNN critic, the derivatives of the QNN actor and critic's k-th entry are as following (5) and (6), respectively,

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_k} = \underbrace{\frac{\partial J(\boldsymbol{\theta})}{\partial \pi_{\boldsymbol{\theta}}} \cdot \frac{\partial \pi_{\boldsymbol{\theta}}}{\partial \langle P_{s,\boldsymbol{\theta}} \rangle}}_{\text{(SGD)}} \cdot \underbrace{\frac{\partial \langle P_{o,\boldsymbol{\theta}} \rangle}{\partial \theta_k}}_{\text{(parameter-shift rule)}}, \tag{5}$$

$$\frac{\partial \mathscr{L}(\boldsymbol{\phi})}{\partial \phi_k} = \underbrace{\frac{\partial \mathscr{L}(\boldsymbol{\phi})}{\partial V_{\boldsymbol{\phi}}} \cdot \frac{\partial V_{\boldsymbol{\phi}}}{\partial \langle P_{s,\boldsymbol{\phi}} \rangle}}_{\text{(SGD)}} \cdot \underbrace{\frac{\partial \langle P_{s,\boldsymbol{\phi}} \rangle}{\partial \phi_k}}_{\text{(parameter-shift rule)}}, \tag{6}$$

where SGD means stochastic gradient descent; (s,a,r,s') and $\mathscr{B}$ denote the current state, current action, reward, next state, and Markov decision process trajectory, respectively. Here, the parameters of QNN actor/critic are represented by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. The first and second derivatives of the right-hand side in (5) and (6) can be computed using classical partial derivative methods, such as gradient descent methods. However, the third derivative cannot be calculated using classical methods, as the quantum state remains unknown prior to measurement. To address this, the parameter-shift rule is adopted to calculate the loss gradient [43]. This parameter-shift rule is applied to (5) for the derivative of the QNN actor's k-th parameter. In addition, it can be determined using the 0-th derivative as follows:

$$\frac{\partial \langle P \rangle_{s,\boldsymbol{\theta}}}{\partial \theta_k} = \langle P \rangle_{s,\boldsymbol{\theta}+\frac{\pi}{2}\mathbf{e}_k} - \langle P \rangle_{s,\boldsymbol{\theta}-\frac{\pi}{2}\mathbf{e}_k}, \tag{7}$$

where $\mathbf{e}_k$ represents the k-th basis of $\boldsymbol{\theta}$. Correspondingly, the left-hand side of (6) can be derived from (7). Similarly, the loss gradient of the QNN critic can be obtained by following the same process as mentioned above. Finally, the gradient of the objective function can be computed as formulated in (2).

D. QRL-BASED SPATIO-TEMPORAL PRIORITIZATION

1) SPATIAL-PRIORITIZATION: QRL-BASED SCHEDULING VIA AVATAR-POPULARITY

This section introduces the spatial-prioritization algorithm that is specifically created for spatial-prioritized synchronization between physical-space and virtual-space using the proposed QRL. It should be noted that the entire process is depicted in Fig. 1.

- *State Space:* The state is defined as the information of the physical-space that the central server (an agent in our RL formulation) observes. The state contains *i)* the environmental information, *ii)* the shared virtual-space information through the server. The server collects the avatar position, the avatar-popularity in virtual-space, the data size of physical-space information, and the age-of-information of physical-space data of all physical space.
- *Action Space:* In spatial-prioritization, the action space is defined as a set of Boolean variables,

  $$\mathscr{A} \triangleq \{a_n\}_{n=1}^{N} \in \{0,1\}^N, \tag{8}$$

  where N stands for the number of virtual/physical space. Note that when $a_n = 1$, the n-th virtual-space is updated with the n-th physical space information. On the other hand, if $a_n = 0$, the virtual-space remains unaltered.
- *Reward Function:* The reward function is designed with the objective and penalty, formulated as follows,

  $$R(s(t),a(t)) = \sum_{n=1}^{N} a_n(t) \cdot \left(\frac{p_n(t)}{q_n(t)} - \lambda \cdot f_c(D_n(t)) \right), \tag{9}$$

  where $p_n(t)$, $q_n(t)$, and $D_n(t)$ denote the avatar popularity, age-of-information, and the accumulated data size in the n-th physical space at time-step t, respectively. Furthermore, $f_c(\cdot)$ represents a quality function as introduced by the authors in [44]. Note that the age-of-information $q_n(t)$ adheres to a time-dependent and non-zero monotonic increasing function.

2) TEMPORAL-PRIORITIZATION: PRIORITIZED VIRTUAL-SPACE RENDERING

This paper aims to adjust the sampling rate of virtual space items, reflecting the user's interest in the updated virtual space information. By doing so, it aims to selectively transmit a reduced amount of data, thereby enhancing our metaverse system.

a: STATE SPACE

First of all, group-popularity is determined according to user preferences, in which each item is assigned a weight, and the sum of these weights is equal to 1, *i.e.*, $\sum_{i=1}^{I} g_i(t) = 1$, $\forall t \in [1,T]$, and $\forall g_i \geq 0$. In order to modify the sampling rate for each item, it is essential to take into account the group popularity. Secondly, we examine the computational stability constraints. As the sampling rate increases, the computational load correspondingly rises. Therefore, we take into consideration a computing queue and aim to prevent divergence. The computing queue dynamics are as follows,

$$Q(t+1) = \max(Q(t) - \sum_{i=1}^{I} b(a_i'(t)), 0) + c(t), \qquad (10)$$

where $a_i'(t)$, $b(a_i'(t))$, and $c(t)$ stand for action at time t, non-zero monotonic decreasing function, and randomly arrived computing tasks, respectively. Finally, both group popularity and the computing queue are subject to fluctuations over time. Given those components from previous time steps constitute vital information, we incorporate current and historical data into the input state.

b: ACTION SPACE

The action space is designed to adjust the sampling rate for each item. Assuming a total of I items and J quality levels, the action space is defined by the sampling rate for each item, *i.e.*,

$$\mathscr{A}' \triangleq \{\{a_i'\}_{i=1}^{I}\} \in \{1/J, 2/J, \cdots, 1.0\}^I, \qquad (11)$$

and this will also be determined via QRL.

c: REWARD FUNCTION

For every item, the sum of the product of group popularity and the sampling ratio is designed as a quality metric that reflects group popularity. The objective is to maximize this quality metric while simultaneously achieving queue stabilization. Finally, the reward function is formulated as,

$$R'(s(t), a'(t)) = \sum_{i=1}^{I} [\nu \cdot g_i(t) \cdot a_i'(t) + Q(t) \cdot b(a_i'(t))], \qquad (12)$$

where ν, $Q(t)$, and $b(\cdot)$ stand for the tradeoff factor between utility and delay, queue backlog at t, and departure process depending on the action taking, which is associated with its quality factor.

Algorithm 1 QRL-Based Scheduler Training

```
Initialize QNN actor and QNN critic parameters ($\boldsymbol{\theta}$ and $\boldsymbol{\phi}$)
for (Epoch = 1, $\cdots$, Training epochs) do
    Initialize Spatio-temporal prioritization environment
    Initialize Replay buffer $\mathscr{B} \leftarrow \{\}$
    for (Time-step = 1, $\cdots$, $T$) do
        Execute Policy $\pi_{\boldsymbol{\theta}}(a(t) \mid s(t))$ to obtain $a(t)$
        Update Spatio-temporal environment
        Store Experience $\xi = \{s(t), a(t), r(t), s(t+1)\}$
        Update Replay buffer $\mathscr{B} \leftarrow \mathscr{B} \cup \xi$
    Execute QNN critic $V_{\boldsymbol{\phi}}$ with replay buffer $\mathscr{B}$
    Calculate Zeroth derivative using (7)
    Update QNN actor network with (2)
    Update QNN critic network with (3)
```

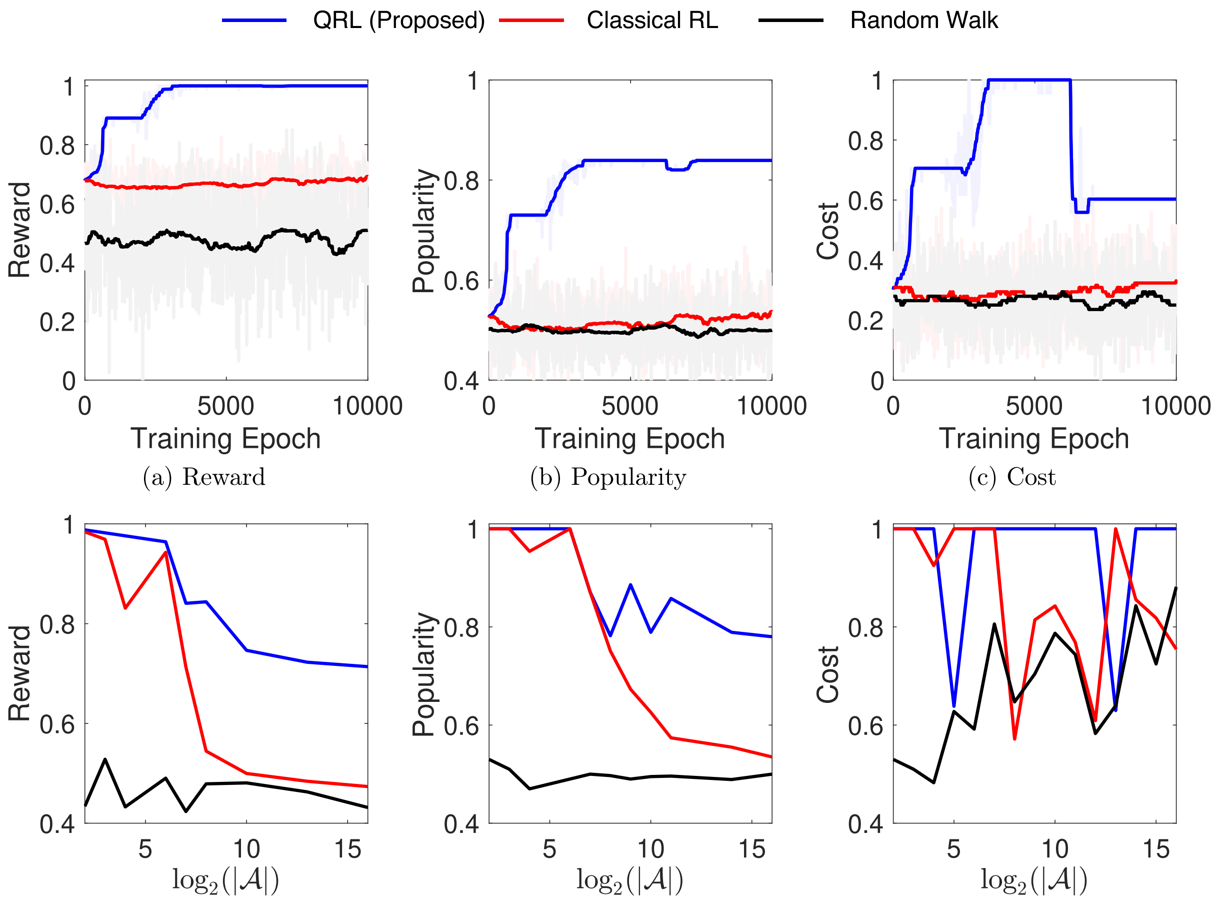
E. PSEUDO-CODE AND COMPUTATIONAL COMPLEXITY

Algorithm 1 represents the comprehensive QRL training procedure for the central server QRL actor-critic networks. First, the weights of the QNN actor and QNN critic networks are established, represented as $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively (line 1). Then, the server continues to train its QNN actor until the conclusion of the training process (lines 2-15). For every training epoch, both the physical-space and virtual-space environments are reset to their initial states, and the server's replay buffer $\mathscr{B}$ is initialized (lines 3-4). The QNN actor, guided by its policy, determines the action $a(t)$ (line 6). Subsequently, the current state $s(t)$ transitions to the succeeding state $s(t+1)$, as dictated by the central server's action decisions (line 7). The transformed environment yields the reward $r(t)$ and transition pairs $\xi = \{s(t), a(t), r(t), s(t+1)\}$, which are stored in the server's experience replay buffer $\mathscr{B}$ (lines 8-9). Once the time step reaches the terminal time T, the QNN critic evaluates the value $V_{\boldsymbol{\phi}}(s)$ for all experiences within the replay buffer $\mathscr{B}$ (line 11). The weights of the QNN actor and QNN critic networks are updated using the parameter-shift rule and the policy gradient (lines 12-14). After completing the training, the central server acquires the optimal QNN actor/critic for spatio-temporal prioritization.

IV. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETTING

This section investigates the superiority of our proposed QRL-based scheduling algorithm in spatio-temporal prioritization. In spatial-prioritization (*i.e.*, Stage #1), QRL-based scheduling aims to maximize the reward function as elaborated on (9), considering the avatar popularity. In temporal-prioritization (*i.e.*, Stage #2), with the spatially optimized virtual space, each algorithm prioritizes virtual-space in temporal to maximize (12). In our metaverse system, the collected data from physical-space are transmitted to the central server. The collected data can be sent to the server with varying levels of quality, represented by QP values of $\{22, 27, 32, 37\}$ [44]. For rendering meta-space, the



(a) Reward (b) Popularity (c) Cost

(d) Reward (e) Popularity (f) Cost

FIGURE 3. **The results of spatial-prioritization: Fig. 3(a–c) illustrate the learning curve of an essential performance metric (control size: 2^{16}), while Fig. 3(d–f) depict the results of the scalability tests.**

TABLE 1. **System parameters for performance evaluation.**

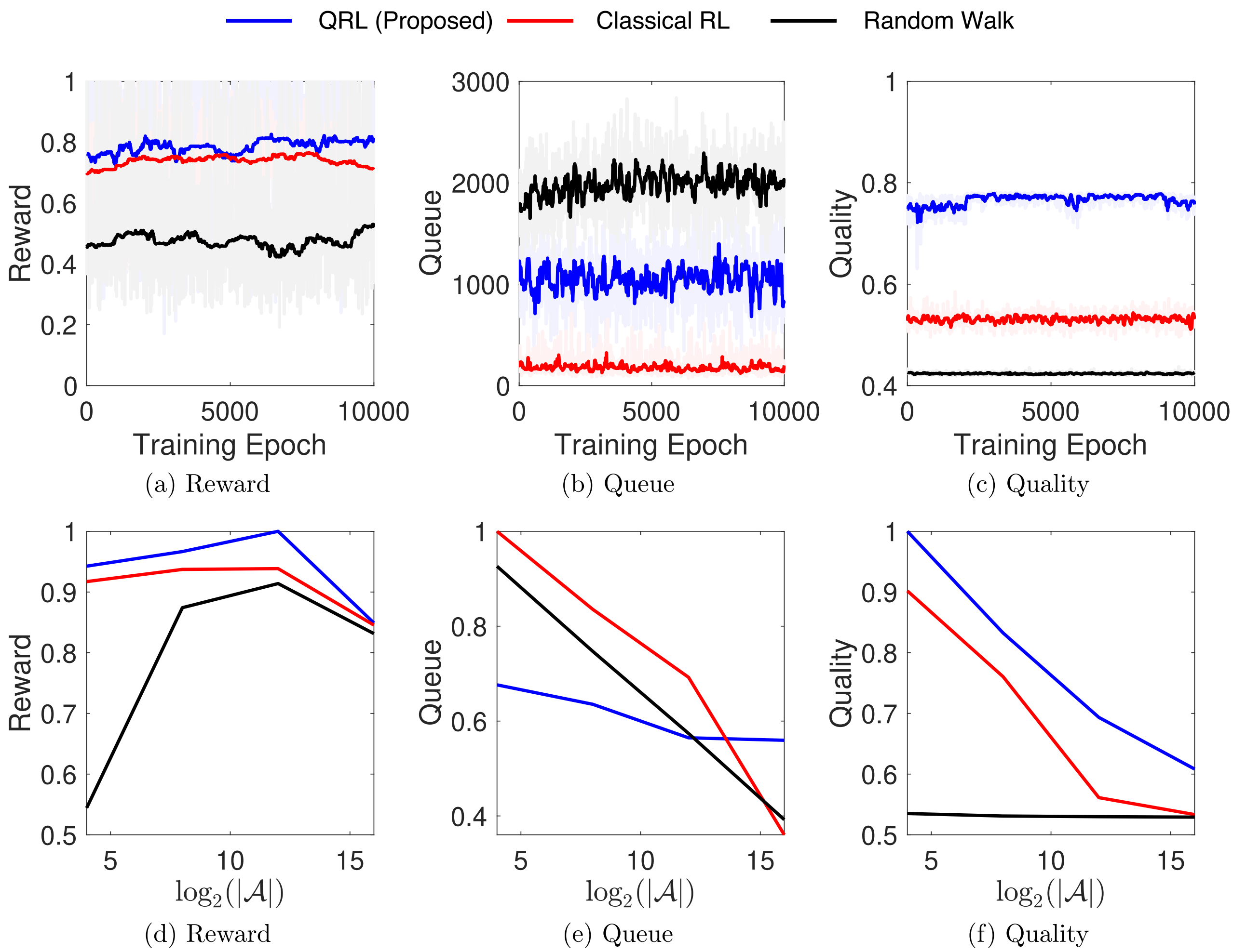
Parameters	Values
The number of physical spaces	$\{1, 2, \cdots, 16\}$
Control dimension ($|\mathscr{A}|$)	$\{2^2, \cdots, 2^{16}\}$
Discount factor (γ)	0.98
Initial value of epsilon	0.5
Minimum epsilon	10^{-2}
Annealing epsilon	5×10^{-5}
Learning rates for actor	10^{-3}
Learning rates for critic	2.5×10^{-4}
Training epochs	10,000
Optimizer	Adam
Hyper-parameters (λ, ν)	(0.01, 100.0)

corresponding bit rates for each QP are 26,496 Kbps, 10,658 Kbps, 5,073 Kbps, and 2,621 Kbps [44]. We use 16 qubit system for spatio-temporal prioritization. The experimental parameters used for performance evaluation are in Table 1.

Benchmarks: To benchmark the proposed algorithm, especially QRL-based scheduling algorithm, the following benchmarks have been conducted. To perform a comparative analysis of quantum computing, a classical RL-based scheduling algorithm, or more precisely, conventional actor-critic networks, are utilized. The primary difference between our QRL and the classical RL lies in the employment of conventional NNs as opposed to QNNs. By employing Monte Carlo simulations, each scheduling participant independently determines their respective actions, irrespective of prevailing states, thus not employing a learning algorithm. Consequently, this allows for the evaluation of performance enhancements achieved through learning by comparing the results with alternative approaches. In addition, we adopt ScanNet point cloud segmentation dataset to visualize the prioritized virtual space [45]. The examples of rendered virtual space in each stage with heterogeneously prioritized user is illustrated in Fig. 5 and Fig. 6, respectively. We further discuss experimental results and their visualization in Sec. IV-D.

B. RESULTS FOR SPATIAL-PRIORITIZATION

Fig. 3 presents the performance evaluation of spatial prioritization. In spatial prioritization, rewards, avatar-popularity, and cost are measured, which are crucial components in objective achievement. Fig. 3(a–c) illustrate the learning



(a) Reward (b) Queue (c) Quality

(d) Reward (e) Queue (f) Quality

FIGURE 4. **The results of temporal-prioritization: Fig. 4(a–c) illustrate the learning curve of an essential performance metric (control size: 2^{16}), while Fig. 4(d–f) depict the results of the scalability tests.**

TABLE 2. **Performance comparison with different algorithm at the end of training (control size by action-dimension: 2^{16}).**

Stage/Metric	Spatial-prioritization			Temporal-prioritization		
	Reward	Popularity	Cost	Reward	Queue	Quality
QRL	1	0.603	0.839	0.814	1085	0.757
RL	0.689	0.331	0.530	0.709	148	0.522
Random walk	0.509	0.25	0.499	0.523	1998	0.423

curve for the three metrics. As shown in Fig. 3(a), the classical RL-based scheduling algorithm and the random walk only achieve 0.67x to 0.48x of the cumulative reward of the QRL-based scheduling algorithm. Furthermore, classical RL-based scheduling algorithm and random walk achieve 0.65x and 0.61x corresponding to the popularity as depicted in Fig. 3(b). On the other hand, the cost of the QRL-based scheduling is higher than comparisons, as shown in Fig. 3(c). However, it is notable that the cost is reduced by 40% compared to the highest value during training. Based on these results, we confirm that our proposed QRL outperforms the comparisons. The detailed values are listed in Table 2. To investigate the effect of control size on the scheduling algorithm, we observe the robustness of our proposed QRL to control size scalability. The results are represented in Fig. 4(d–f). As the control size increases from 2^1 to 2^{16}, performance degrades up to 29% due to limited resources. On the other hand, the performance of the classical RL-based scheduling algorithm in the large control size is degraded by 56% compared to the small control size. This is because the efficacy of traditional RL-based scheduling algorithms can be significantly deteriorated by the curse of dimensionality [39],

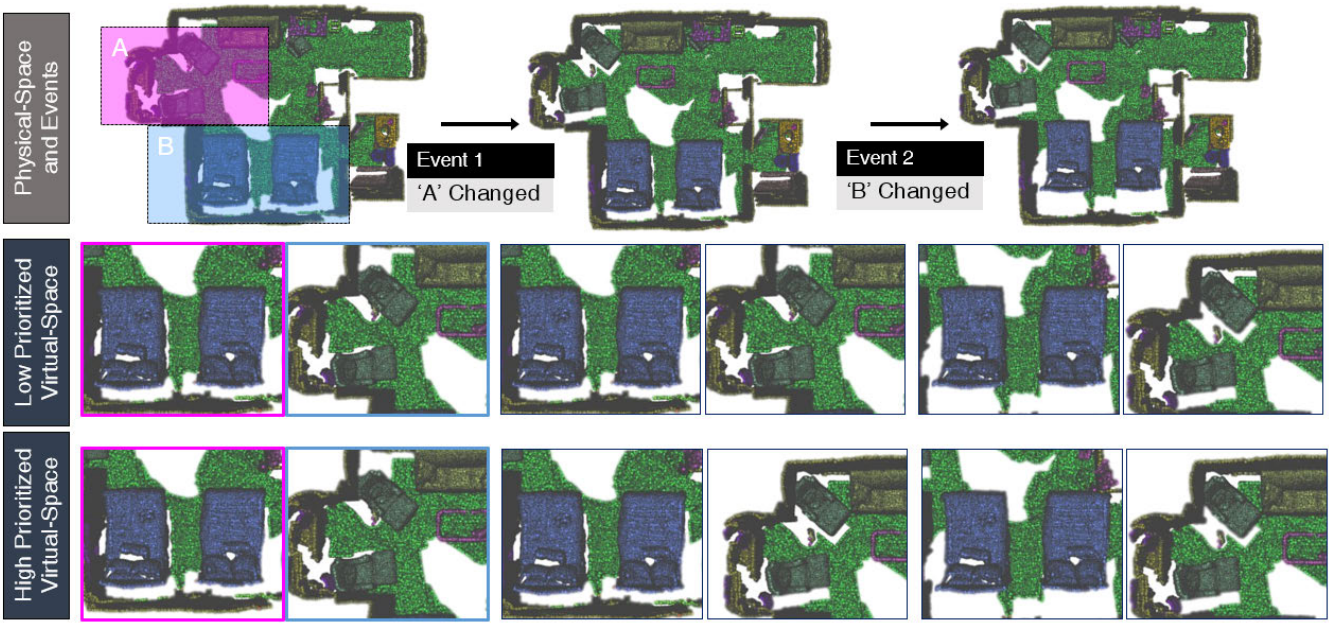

Physical-Space and Events
A
B
Event 1
'A' Changed
Event 2
'B' Changed
Low Prioritized Virtual-Space
High Prioritized Virtual-Space

FIGURE 5. Spatial-prioritization results on ScanNet point cloud segmentation dataset [45].

given that the control size in spatial prioritization grows exponentially as the number of regions increases.

C. RESULTS FOR TEMPORAL-PRIORITIZATION

Fig. 4 presents the performance evaluation of temporal prioritization. Similar to the experiments conducted for spatial prioritization, temporal prioritization also assesses the essential factors contributing to objective achievement. Fig. 4(a–c) show the learning curve for the three metrics. As illustrated in Fig. 4(a), the QRL-based scheduling algorithm achieves the highest performance, followed by classical RL-based scheduling algorithm and random walk. At the end of the training, the cumulative rewards for each algorithm are 0.814, 0.709, and 0.523 for QRL, classical RL, and random walk, respectively. From the computing queue perspective, the random walk's computing queue overflows, while the classical RL's computing queue remains empty, indicating low sampling rates. In contrast, our proposed QRL-based scheduling algorithm stabilizes the computing queue at a value of 1085. This result is confirmed in Fig. 4(b). Fig. 4(c) demonstrates that the quality metric of our proposed QRL algorithm exceeds the other comparisons. The detailed values are listed in Table 2. Fig. 4(d–f) depict the performance evaluation metrics for various control sizes. Fig. 4(d) presents the reward metric for various control sizes. When the control size is small, specifically 2^2, there is no noticeable difference between the QRL-based scheduling algorithm and the classical RL-based scheduling algorithm. However, the QRL-based scheduling algorithm significantly outperforms the classical RL with a value of 1.09x when the control size increases, specifically to 2^{16}. Similar trends are observed in cost and quality, as shown in Fig. 4(e)/(f). The largest performance gap between QRL and other comparisons occurs when the action dimension is 2^{16}. The gap amounts to 0.086, 0.195, and 0.075 for reward, queue, and quality, respectively.

In summary, experiments of temporal prioritization validate that our QRL-based scheduling achieves a performance improvement of up to 9%, particularly when the action dimension increases.

D. VISUALIZATION

Along with the above experimental results, Fig. 5 visualizes a brief example of the QRL-based spatial prioritization process. Region A and region B represent physical regions with high and low avatar-popularity, respectively. To alleviate the constraints of limited communication and computing resources, synchronization between physical and low-prioritized virtual spaces occurs only when there are changes in the high avatar-popularity region, whereas high-prioritized virtual spaces synchronize whenever changes occur. Fig. 6 illustrates the process of temporal prioritization. After the spatial prioritization carried out in Stage #1, the synchronized virtual space undergoes the sequential process based on group popularity. To account for the heterogeneity in group popularity, each object group within the prioritized virtual space is rendered using different sampling ratios. Group-popularity is represented by the corresponding colors (i.e., labels) assigned to objects in the virtual space. For example, when the group-popularity of group 3 is high on beds and floors, the sampling ratio of the other objects is decreased, resulting in an overall performance increase while maintaining the quality of user services.

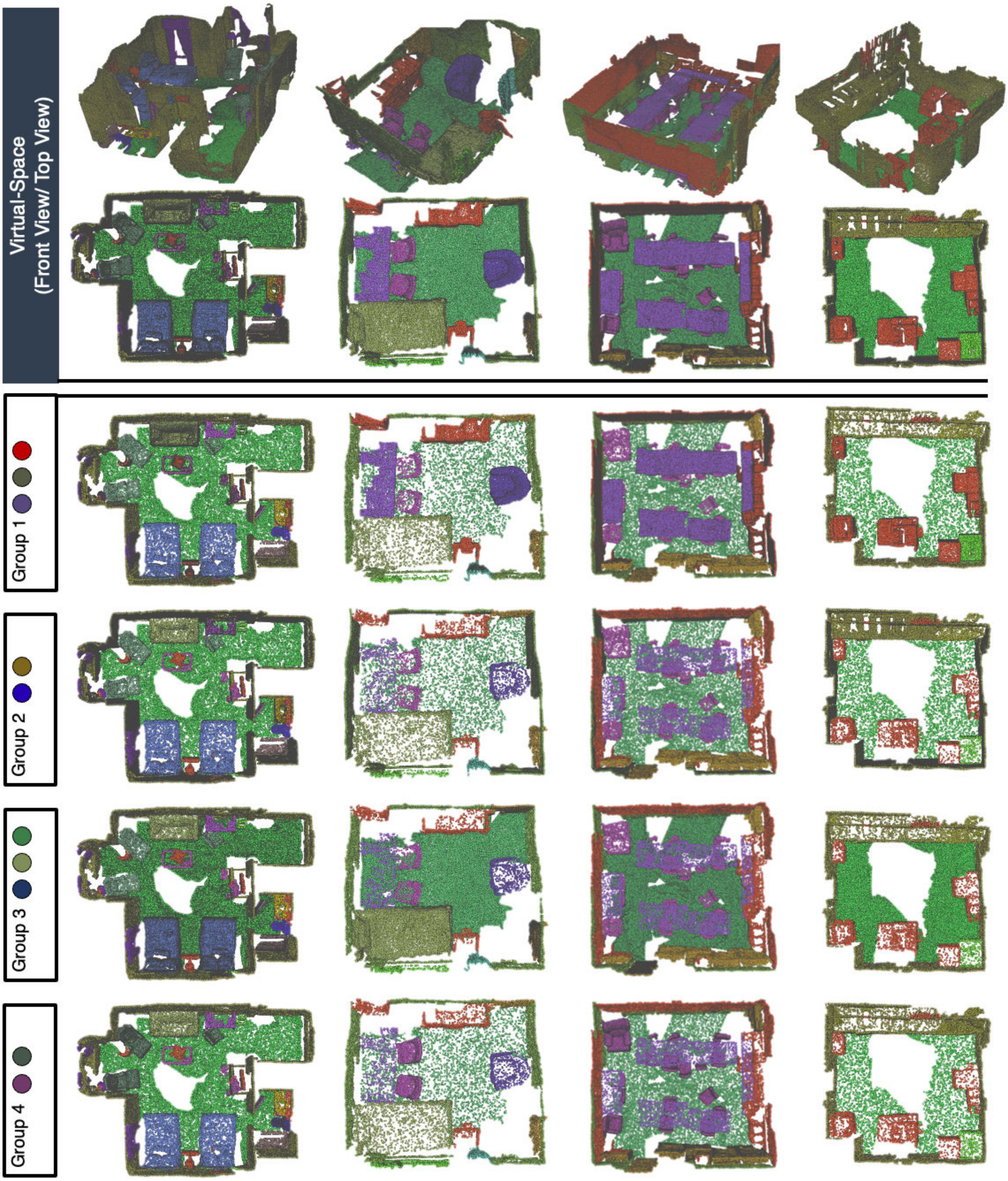


FIGURE 6. Temporal prioritization results on ScanNet point segmentation dataset. Each column represents different data points, and each group has different group-popularity.

In summary, we observe that our QRL-based scheduling in temporal prioritization demonstrates feasible performance in accurately reflecting group-popularity within the virtual-space.

V. CONCLUDING REMARKS AND FUTURE WORK

Considering the importance of synchronization between physical-space and virtual-space under the consideration of limited communication and computing resources for realistic metaverse services, a novel spatio-temporal prioritization is proposed in this paper using high-performance quantum computing techniques. The proposed prioritization algorithm comprises two stages. The first stage (*i.e.*, spatial prioritization) aims for low-delay data gathering from the physical-space to construct the realistic virtual-space. The second stage (*i.e.*, temporal prioritization) involves rendering items at various sampling rates, taking them into account to maximize high-reality rendering quality in virtual-space under resource limitations. According to the large control size of spatio-temporal prioritization, classical deep learning algorithms faces challenges due to the curse of dimensionality. To tackle this problem, we propose a QRL-based scheduling algorithm aims at the action space dimension reduction in spatio-temporal prioritization. Our proposed QRL-based scheduling algorithm with the logarithmic-scale of qubit utilization reduction can be realized, which is crucial in the NISQ era. By employing the QRL-based scheduling algorithm in this paper, we confirm that our proposed spatio-temporal prioritization can be efficiently utilized due to logarithmic-sacle action dimension reduction. In addition, we conduct extensive experiments by constructing realistic metaverse systems using the ScanNet dataset, which leverages point clouds for realistic representation. With these experiments, we corroborate that our proposed algorithm achieves superiority over other classical prioritization algorithms.

As future work, it is considerable to design a new QRL algorithms for various metaverse applications. In modern multimedia research, the system exists where the virtual-space should be enhanced subject to physical-space constraints, e.g., digital-twin systems. In digital-twin, the secondary virtual images should be constructed and rendered subject to network and original image constraints. Therfore, it is possible to consdier to utilize QRL-based algorithms in digital-twin networks.

REFERENCES

[1] G. Bansal, K. Rajgopal, V. Chamola, Z. Xiong, and D. Niyato, "Healthcare in metaverse: A survey on current metaverse applications in healthcare," *IEEE Access*, vol. 10, pp. 119914–119946, 2022.

[2] C. Latulipe, "A CS1 team-based learning space in gather.town," in *Proc. 52nd ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2021, p. 1245.

[3] J. D. N. Dionisio, W. G. B. Iii, and R. Gilbert, "3D virtual worlds and the metaverse: Current status and future possibilities," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–38, Jun. 2013.

[4] W.-Z. Nie, W.-W. Jia, W.-H. Li, A.-A. Liu, and S.-C. Zhao, "3D pose estimation based on reinforce learning for 2D image-based 3D model retrieval," *IEEE Trans. Multimedia*, vol. 23, pp. 1021–1034, 2021.

[5] J. Wang, H. Du, Z. Tian, D. Niyato, J. Kang, and X. Shen, "Semantic-aware sensing information transmission for metaverse: A contest theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5214–5228, Aug. 2013.

[6] X. Lin, K. Chen, and K. Jia, "Object point cloud classification via poly-convolutional architecture search," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 807–815.

[7] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 153–161.

[8] T. M. Khan and A. Robles-Kelly, "Machine learning: Quantum vs classical," *IEEE Access*, vol. 8, pp. 219275–219294, 2020.

[9] M. Lombardi and M. Milano, "Boosting combinatorial problem modeling with machine learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 5472–5478.

[10] Y. Kwak, W. J. Yun, J. P. Kim, H. Cho, J. Park, M. Choi, S. Jung, and J. Kim, "Quantum distributed deep learning architectures: Models, discussions, and applications," *ICT Exp.*, vol. 9, no. 3, pp. 486–491, Jun. 2023.

[11] W. J. Yun, J. P. Kim, H. Baek, S. Jung, J. Park, M. Bennis, and J. Kim, "Quantum federated learning with entanglement controlled circuits and superposition coding," 2022, *arXiv:2212.01732*.

[12] E. Pelofske, A. Bartschi, and S. J. Eidenbenz, "Quantum volume in practice: What users can expect from NISQ devices," 2022, *arXiv:2203.03816*.

[13] T. Huang, R.-X. Zhang, and L. Sun, "Zwei: A self-play reinforcement learning framework for video transmission services," *IEEE Trans. Multimedia*, vol. 24, pp. 1350–1365, 2022.

[14] S. Park, J. P. Kim, C. Park, S. Jung, and J. Kim, "Quantum multi-agent reinforcement learning for autonomous mobility cooperation," *IEEE Commun. Mag.*, early access, Aug. 28, 2023, doi: 10.1109/MCOM.020.2300199.

[15] L. Cui, D. Su, S. Yang, Z. Wang, and Z. Ming, "TCLiVi: Transmission control in live video streaming based on deep reinforcement learning," *IEEE Trans. Multimedia*, vol. 23, pp. 651–663, 2021.

[16] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9895–9903, Oct. 2020.

[17] H. Baek, S. Park, and J. Kim, "Logarithmic dimension reduction for quantum neural networks," in *Proc. ACM Conf. Inf. Knowl. Manag. (CIKM)*, Birmingham, U.K., Oct. 2023, pp. 3738–3742.

[18] C. O. Jaynes, W. B. Seales, K. L. Calvert, Z. Fei, and J. Griffioen, "The metaverse—A networked collection of inexpensive, self-configuring, immersive environments," in *Proc. Int. Workshop Immersive Projection Technol. (IPT)*, Zurich, Switzerland, May 2003, pp. 115–124.

[19] X. Chen, W. Liu, X. Liu, Y. Zhang, J. Han, and T. Mei, "MAPLE: Masked pseudo-labeling autoencoder for semi-supervised point cloud action recognition," in *Proc. ACM Int. Conf. Multimedia (MM)*, Lisboa, Portugal, Oct. 2022, pp. 708–718.

[20] F. Shao, Y. Luo, P. Liu, J. Chen, Y. Yang, Y. Lu, and J. Xiao, "Active learning for point cloud semantic segmentation via spatial-structural diversity reasoning," in *Proc. 30th ACM Int. Conf. Multimedia*, Lisboa, Portugal, 2022, pp. 2575–2585.

[21] W. Sun, P. Wang, N. Xu, G. Wang, and Y. Zhang, "Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5839–5852, Apr. 2022.

[22] H. Zhang, G. Zhang, and Q. Yan, "Dynamic resource allocation optimization for digital twin-driven smart shopfloor," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Zhuhai, China, Mar. 2018, pp. 1–5.

[23] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4968–4977, Jul. 2021.

[24] K. Rohmer and T. Grosch, "Tiled frustum culling for differential rendering on mobile devices," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, Fukuoka, Japan, Sep. 2015, pp. 37–42.

[25] A. Kumari, D. Vekaria, R. Gupta, and S. Tanwar, "Redills: Deep learning-based secure data analytic framework for smart grid systems," in *Proc. IEEE Int. Conf. Commun. (ICC) Workshops*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[26] S. Tanwar, A. Kumari, D. Vekaria, M. S. Raboaca, F. Alqahtani, A. Tolba, B.-C. Neagu, and R. Sharma, "GrAb: A deep learning-based data-driven analytics scheme for energy theft detection," *Sensors*, vol. 22, no. 11, p. 4048, May 2022.

[27] A. Kumari, M. Trivedi, S. Tanwar, G. Sharma, and R. Sharma, "SV2G-ET: A secure vehicle-to-grid energy trading scheme using deep reinforcement learning," *Int. Trans. Electr. Energy Syst.*, vol. 2022, pp. 1–11, Apr. 2022, Art. no. 9761157.

[28] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 319–352, 1st Quart., 2023.

[29] S.-M. Park and Y.-G. Kim, "A metaverse: Taxonomy, components, applications, and open challenges," *IEEE Access*, vol. 10, pp. 4209–4251, 2022.

[30] H. Huang, X. Zeng, L. Zhao, C. Qiu, H. Wu, and L. Fan, "Fusion of building information modeling and blockchain for metaverse: A survey," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 195–207, 2022.

[31] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang, and Z. Zheng, "Fusing blockchain and AI with metaverse: A survey," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 122–136, 2022.

[32] Y. Lin, H. Du, D. Niyato, J. Nie, J. Zhang, Y. Cheng, and Z. Yang, "Blockchain-aided secure semantic communication for AI-generated content in metaverse," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 72–83, 2023.

[33] D. M. Doe, J. Li, N. Dusit, Z. Gao, J. Li, and Z. Han, "Promoting the sustainability of blockchain in Web 3.0 and the metaverse through diversified incentive mechanism design," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 171–184, 2023.

[34] Z. Meng, C. She, G. Zhao, and D. De Martini, "Sampling, communication, and prediction co-design for synchronizing the real-world device and digital model in metaverse," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 288–300, Jan. 2023.

[35] J. Wu, K. Lin, D. Lin, Z. Zheng, H. Huang, and Z. Zheng, "Financial crimes in Web3-empowered metaverse: Taxonomy, countermeasures, and opportunities," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 37–49, 2023.

[36] U. Chabaud, D. Markham, and A. Sohbi, "Quantum machine learning with adaptive linear optics," *Quantum*, vol. 5, p. 496, Jul. 2021.

[37] W. J. Yun, S. Park, J. Kim, M. Shin, S. Jung, D. A. Mohaisen, and J.-H. Kim, "Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-UAV control," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7086–7096, Oct. 2022.

[38] W. Yun, Y. Kwak, J. Kim, H. Cho, S. Jung, J. Park, and J. Kim, "Quantum multi-agent reinforcement learning via variational quantum circuit design," in *Proc. IEEE ICDCS*, Jul. 2022, pp. 1332–1335.

[39] W. Du and S. Ding, "A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications," *Artif. Intell. Rev.*, vol. 54, no. 5, pp. 3215–3238, Jun. 2021.

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

[41] C. Park, W. J. Yun, J. P. Kim, T. K. Rodrigues, S. Park, S. Jung, and J. Kim, "Quantum multi-agent actor-critic networks for cooperative mobile access in multi-UAV systems," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 20033–20048, Nov. 2023.

[42] S. Jerbi, C. Gyurik, S. Marshall, H. J. Briegel, and V. Dunjko, "Variational quantum policies for reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2021, pp. 28362–28375.

[43] G. E. Crooks, "Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition," 2019, *arXiv:1905.13311*.

[44] J. Kim and E.-S. Ryu, "Feasibility study of stochastic streaming with 4K UHD video traces," in *Proc. IEEE Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju Island, (South) Korea, Oct. 2015, pp. 1350–1355.

[45] A. Dai, M. Nießner, M. Zollhofer, S. Izadi, and C. Theobalt, "Bundle-Fusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration," 2016, *arXiv:1604.01093*.

SOOHYUN PARK received the B.S. degree in computer science and engineering from Chung-Ang University, Seoul, South Korea, in February 2019, and the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, Korea University, Seoul, in August 2023.

She was a Postdoctoral Scholar with the Department of Electrical and Computer Engineering, Korea University, from September 2023 to February 2024. She has been an Assistant Professor with Sookmyung Women's University, Seoul, since March 2024. Her research interests include deep learning theory and network/mobility applications, quantum neural network (QNN) theory and applications, QNN software engineering and programming languages, and AI-based autonomous control for distributed computing systems. She was a recipient of the ICT Express Best Reviewer Award (2021), the IEEE Seoul Section Student Paper Contest Award, and the IEEE Vehicular Technology Society (VTS) Seoul Chapter Award.

HANKYUL BAEK received the B.S. degree in electrical engineering from Korea University, Seoul, Republic of Korea, in February 2020, and the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, Korea University, in February 2024.

He was with LG Electronics, Seoul, from 2020 to 2021. He was a Visiting Scholar with the Department of Electrical and Computer Engineering, The University of Utah, Salt Lake City, UT, USA, in 2023. He has been a Postdoctoral Scholar with the Department of Electrical and Computer Engineering, Korea University, since March 2024. His current research interests include quantum machine learning and its applications.

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

He has been with Korea University, since 2019, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering and also an Adjunct Professor with the Department of Communications Engineering (co-operated by Samsung Electronics) and the Department of Semiconductor Engineering (co-operated by SK Hynix). Before joining Korea University, he was a Research Engineer with LG Electronics, Seoul, 2006–2009, a Systems Engineer with Intel Corporation Headquarter, Santa Clara, CA, USA, 2013–2016, and an Assistant Professor of computer science and engineering with Chung-Ang University, Seoul, 2016–2019. He was a recipient of the Annenberg Graduate Fellowship with his Ph.D. admission from USC (2009), the Intel Corporation Next Generation and Standards (NGS) Division Recognition Award (2015), the IEEE Systems Journal Best Paper Award (2020), the IEEE ComSoc Multimedia Communications Technical Committee (MMTC) Outstanding Young Researcher Award (2020), and the IEEE ComSoc MMTC Best Journal Paper Award (2021). He also received several awards from IEEE conferences, including the IEEE ICOIN Best Paper Award (2021), the IEEE ICTC Best Paper Award (2022), and the IEEE Vehicular Technology Society (VTS) Seoul Chapter Award. He serves as an Editor and the Guest Editor for IEEE Transactions on Vehicular Technology, IEEE Internet of Things Journal, *IEEE Communications Standards Magazine*, *Computer Networks* (Elsevier), and *ICT Express* (Elsevier).