2026 International Conference on Quantum Communications, Networking, and Computing (QCNC)

Effective Entanglement Scheduling in Quantum Networks via Deep Reinforcement Learning

Xun Wang
IIIS
Tsinghua University
Beijing, China
wang-x24@mails.tsinghua.edu.cn

Longbo Huang*
IIIS
Tsinghua University
Beijing, China
longbohuang@tsinghua.edu.cn

John C.S. Lui*
CSE
The Chinese University of Hong Kong
Shatin, N.T, Hong Kong
cslui@cse.cuhk.edu.hk

2026 International Conference on Quantum Communications, Networking, and Computing (QCNC) | 979-8-3315-6110-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/QCNC69040.2026.00073

***Abstract*—Quantum entanglement is the cornerstone of quantum network communication, yet the qubit resources required to establish it are severely limited. Moreover, users within the network impose strict delay constraints on entanglement requests due to various factors, such as the real-time requirements of upper-layer applications or the finite coherence time of user-side quantum memories. Consequently, effectively scheduling scarce qubit resources to serve these time-sensitive requests presents a significant challenge. This difficulty is exacerbated by the partial observability of quantum systems and the time-varying nature of qubit generation rates, rendering traditional scheduling algorithms ineffective. Furthermore, while Deep Reinforcement Learning (DRL) offers a promising avenue for quantum network optimization, existing research primarily focuses on network-layer routing, overlooking the finer-grained entanglement scheduling problem subject to strict delay and resource constraints. To bridge this gap, we propose the Deep Quantum Request Scheduler (DQRS), a novel DRL-based algorithm designed to maximize service quality under these dual constraints. Specifically, DQRS integrates a spatio-temporal branching architecture and virtual action slots within an Actor-Critic framework. This design enables robust scheduling with continuous actions while adaptively managing fluctuating qubit budgets. Extensive simulation results demonstrate that DQRS achieves up to 44% performance improvement over traditional methods and a 4% gain compared to state-of-the-art DRL benchmarks across various resource constraints and observability settings, validating the robustness of the proposed approach.**

***Index Terms*—Quantum networks, Entanglement scheduling, Deep reinforcement learning.**

I. Introduction

The realization of quantum networks relies heavily on entanglement distribution [1]–[3], which provides the essential resources for protocols such as quantum teleportation, secure cryptography, and distributed quantum computing. However, serving these applications presents a fundamental challenge: the network must satisfy strict delay constraints imposed by users while operating with limited qubit resources. These stringent timing requirements stem from the specific Quality of Service demands of end-users. For instance, many upper-layer applications necessitate high real-time responsiveness to ensure the seamless execution of interactive quantum protocols. Meanwhile, the inevitable decoherence of user-side quantum memories imposes intrinsic delay constraints on critical operations, such as entanglement swapping, where the scheduler must promptly deliver fresh entanglement before the fidelity of existing states degrades below the required threshold. Crucially, if the scheduler fails to serve a request within its critical validity window, the request becomes invalid, leading to immediate service failure. This not only severely impairs the system throughput but also results in the wastage of scarce quantum resources already committed to the failed task.

The challenge is further intensified by the stochastic and time-varying nature of qubit generation rates, which requires the scheduler to allocate resources under dynamic, time-dependent budgets intelligently. Adding to this complexity is the partial observability of the quantum system, where the complete network state is often inaccessible, introducing high uncertainty into the decision-making process. Consequently, developing an efficient entanglement scheduling algorithm capable of managing these strict trade-offs is of paramount importance for optimizing quantum network performance.

However, traditional scheduling algorithms [4]–[8] often struggle to cope with the partial observability of quantum network states and fail to meet the strict delay and resource constraints. In contrast, Deep Reinforcement Learning (DRL) has emerged as a powerful paradigm for solving complex decision-making problems, particularly in environments characterized by uncertainty and partial information. By learning optimal policies through continuous interaction with the environment, DRL eliminates the reliance on precise system models. Leveraging these inherent advantages, DRL has recently garnered increasing attention and has been widely adopted in the context of quantum entanglement distribution [1], [9]–[11].

Nevertheless, existing DRL-based approaches primarily focus on network-layer routing optimization, often overlooking the finer-grained entanglement request scheduling problem subject to strict delay and resource constraints, which presents more significant challenges. First, in contrast to routing tasks that often rely on macroscopic network metrics, fine-grained scheduling must contend with intricate mechanisms, such as strict delay constraints and the stochastic probabilities

The work of Xun Wang and Longbo Huang was supported by the National Natural Science Foundation of China Grant 52494974. The work of John C.S. Lui was supported in part by the RGC SRFS2122-4S02 and GRF-14202925.

*Corresponding authors.

979-8-3315-6110-9/26/$31.00 ©2026 IEEE
DOI 10.1109/QCNC69040.2026.00073

of entanglement establishment. This necessitates a level of temporal precision and responsiveness to micro-state dynamics that is rarely demanded in high-level routing optimization. Second, practical quantum systems are subject to time-varying qubit generation rates, where the available qubit budget fluctuates dynamically over time. This temporal variability renders simple greedy strategies ineffective, compelling the scheduler to strategically manage qubit consumption to accommodate future scarcity. Finally, unlike routing decisions that typically involve discrete action sets (e.g., path selection), optimizing request scheduling under these complex dynamics often requires operating in a continuous action space to allocate qubit resources. This substantially expands the search range and complicates policy convergence compared to traditional discrete control problems.

In this paper, to address these challenges, we model the delay-critical entanglement distribution process under dynamic qubit budgets as a delay-constrained scheduling problem [12]–[16], explicitly incorporating time-varying resource availability [8]. Subsequently, we formulate the underlying system dynamics as a Partially Observable Markov Decision Process and propose a novel DRL algorithm, named **D**eep **Q**uantum **R**equest **S**cheduler (DQRS), to derive the optimal policy. Specifically, DQRS enhances policy learning by leveraging a spatio-temporal branching architecture for feature extraction, incorporating a virtual action slot mechanism to guarantee resource causality, and employing a robust Actor-Critic paradigm to facilitate stable policy optimization in continuous action spaces. Extensive simulation results demonstrate that under various resource constraints and observation settings, DQRS achieves a performance gain of up to **44%** over traditional methods and up to **4%** over the state-of-the-art DRL baseline, underscoring its effectiveness.

II. Related Work

A. Quantum Network Optimization

A substantial body of research focuses on optimizing entanglement routing paths. For instance, prior studies have investigated diverse routing protocols [17]–[21]. Furthermore, specialized architectural solutions have been explored [22]–[24]. Attention has also been devoted to security assessment [8], multi-party entanglement support [25], and multi-source distribution strategies [26].

Complementing path selection, optimization-based resource allocation plays a pivotal role. Regarding network stability, researchers have applied Lyapunov drift optimization [4] and max-weight policies [5], or addressed utility maximization under adversarial bandit feedback [6]. Parallel efforts focus on optimizing specific performance metrics, including throughput, fidelity, and fairness [7], [27], [28]. In addition, theoretical analyses have characterized capacity regions under purification [29] and evaluated the switching performance of distribution policies [30].

Nevertheless, most prior studies either overlook the simultaneous impact of resource limits and strict delay constraints, or rely on prior knowledge that is unavailable in practical quantum network systems.

B. Deep Reinforcement Learning for Quantum Network

Recently, Deep Reinforcement Learning (DRL) has emerged as a powerful paradigm for complex decision-making under uncertainty [31]–[37]. However, in the context of quantum networking, existing DRL-based efforts have predominantly concentrated on network-layer routing optimization [1], [9]–[11], leaving the finer-grained entanglement request scheduling problem, subject to the dual constraints of strict deadlines and limited qubit resources, largely unexplored. While analogous delay-constrained scheduling problems have been investigated in classical networking [14]–[16], these works typically rely on the assumption of static or average resource budgets. Such assumptions are ill-suited for quantum scenarios, where the available qubit budget is inherently time-varying. Since users consume finite qubit resources to establish fresh entanglement pairs, the dynamic fluctuation of qubit availability necessitates more adaptive and causality-aware control policies tailored to these unique resource dynamics.

III. Problem Formulation

A. Entanglement Scheduling

Inspired by prior research [12]–[16], we consider a discrete-time quantum network architecture with a star topology, where a central scheduler serves N directly connected users, indexed by $i = 1, 2, \ldots, N$. Time is slotted and denoted by $t = 1, 2, \ldots, T$. In each time slot t, entanglement requests are generated stochastically based on the users' demands. Let $A_i(t)$ denote the number of entanglement requests from user i in slot t. The arrival vector is defined as $\mathbf{A}(t) = [A_1(t), A_2(t), \ldots, A_N(t)]$. The scheduler then determines how to allocate limited qubit resources to generate new entangled pairs and serve these users.

1) Delay Constraints: The entanglement requests have rigid deadlines stemming from various factors, such as the timeliness requirements of upper-layer applications or the decoherence effects at the link layer. To model this time-sensitivity, we define τ_i as the maximum allowable delay (in time slots) for the requests of user i. Consequently, any request that is not successfully served within τ_i slots of its generation is considered expired and is immediately purged from the queue. This mechanism prevents the futile consumption of scarce qubit resources on stale requests that can no longer satisfy the user's requirements.

2) Buffer Model: The central scheduler maintains N separate logical queues to buffer the entanglement requests. Let the aggregate queue state of the network be denoted as:

$$\mathbf{Q}(t) = [Q_1(t), Q_2(t), \ldots, Q_N(t)]. \tag{1}$$

The i-th queue, $Q_i(t)$, buffers the pending requests associated with user i. To explicitly track the urgency of these delay-critical requests, each queue is structured as a set of sub-queues based on the residual lifetime: $Q_i(t) = [Q_i^0(t), Q_i^1(t), \ldots, Q_i^{\tau_i}(t)]$. Here, $Q_i^j(t)$ stores the number of

requests for user i that have exactly j time slots remaining before the deadline. Note that requests with $j = 0$ represent the tasks that will expire and be removed from the system at the beginning of the next time slot if not served successfully.

3) Entanglement Generation: To serve a request in slot t, the scheduler must allocate qubit resources to establish fresh entangled pairs for service. Due to the probabilistic nature of photon transmission and entanglement generation, a single attempt often yields a low success rate. To mitigate this, we assume the system adopts a parallel trial strategy (e.g., spatial multiplexing), where multiple qubits are consumed simultaneously to attempt entanglement generation for a single request [38], [39]. Let $\boldsymbol{\nu}_i(t) = [\nu_i^0(t), \nu_i^1(t), \ldots, \nu_i^{\tau_i}(t)]$ denote the qubit allocation vector for user i, where $\nu_i^j(t) \in [0, \nu_{\max}]$ represents the number of qubits allocated per request within the j-th sub-queue. Here, $\nu_{\max}$ denotes the hardware capacity limit, representing the maximum number of parallel quantum memories or channels available for a single link interface. For the purpose of optimization, we model ν as a continuous real variable representing the resource activation intensity, facilitating gradient-based policy learning.

The success probability of an entanglement request is contingent upon the quantity of qubit resources allocated for its establishment and the prevailing channel conditions during the distribution process. We denote the channel state information for all users at time t as $\mathbf{c}(t) = [c_1(t), c_2(t), \ldots, c_N(t)]$. Consequently, the service attempt for each individual request of user i is modeled as a resource-dependent Bernoulli trial. Specifically, all $Q_i^j(t)$ requests within the same sub-queue share the same success probability, denoted as $p_i(\nu, c_i(t))$. If a service attempt fails (with probability $1 - p_i$), the request remains in the queue for the next slot, provided its residual lifetime $j > 0$.

This probability is monotonically increasing with respect to the allocated qubit count ν (benefiting from diversity gain via parallel trials), while being dynamically influenced by the instantaneous channel state $c_i(t)$. However, the exact analytical relationship governing these dependencies remains unknown to the scheduler.

4) Dynamic Qubit Resource Model: Accounting for the stochastic nature of qubit resource availability, we adopt the time-varying resource constraint model framework from [8]. We define $B(t)$ as the available qubit budget at the beginning of time slot t, with an initial level $B(0) = B_{\text{init}}$. The system operates under a dynamic supply process, where a stochastic amount of fresh qubit resources, denoted as $G(t)$, becomes available for use in each time slot. To maintain consistency with the continuous action space formulation defined in the previous section, we model $G(t)$ as a continuous random variable following an Exponential distribution with mean λ. This effectively captures the random fluctuations in qubit supply intensity.

The total resource consumption by the scheduler at time slot t is denoted as $N_{\text{cons}}(t)$. This consumption is determined by the scheduling policy and is given by the summation of allocated resources across all users:

$$N_{\text{cons}}(t) = \sum_{i=1}^{N} Q_i(t)^\top \boldsymbol{\nu}_i(t). \tag{2}$$

The system is subject to the resource causality constraint, which requires that the consumed resources cannot exceed the currently available budget, i.e., $N_{\text{cons}}(t) \leq B(t)$. Consequently, assuming the node possesses sufficient quantum memory capacity, the dynamics of the available qubit budget evolve according to:

$$B(t+1) = B(t) - N_{\text{cons}}(t) + G(t). \tag{3}$$

5) Scheduler Objective: Let $u_i(t)$ denote the number of successfully served entanglement requests for user i in time slot t. Each user is assigned a predefined priority weight w_i to reflect distinct service requirements (e.g., varying importance or application criticalities). Consequently, the instantaneous weighted throughput of the system, denoted as $U(t)$, and the long-term time-average weighted throughput, denoted as $\overline{U}$, are defined as:

$$U(t) = \sum_{i=1}^{N} w_i u_i(t), \quad \overline{U} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} U(t). \tag{4}$$

The objective of the scheduler is to maximize the long-term time-average weighted throughput, subject to the strict qubit resource causality constraint in each time slot. The optimization problem is formulated as follows:

$$\begin{aligned} \mathcal{P}: \max_{\boldsymbol{\nu}_i(t):\ 1\leq i\leq N} \quad & \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i u_i(t) \\ \text{s.t.} \quad & \sum_{i=1}^{N} \mathbf{Q}_i(t)^\top \boldsymbol{\nu}_i(t) \leq B(t), \quad \forall t \\ & \nu_i^j(t) \in [0, \nu_{\max}], \quad \forall i, j, t. \end{aligned} \tag{5}$$

In practical quantum network systems, the scheduler typically lacks prior knowledge of system dynamics, such as the stochastic arrival rates of requests and qubit resources, the explicit functional form of the entanglement success probability, and the transition probabilities governing channel evolution. As a result, it is intractable to directly solve the aforementioned constrained optimization problem using traditional methods. To address this challenge, we reformulate the problem as a Partially Observable Markov Decision Process (POMDP) and design a novel Deep Reinforcement Learning algorithm to approximate the optimal policy.

B. Partially Observable Markov Decision Process

A Partially Observable Markov Decision Process (POMDP) [40] is characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}, r, \mathcal{O}, \Omega, \gamma, \mu_0 \rangle$, where $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{O}$ denote the state space, action space, and observation space, respectively. r is the reward function, and $\mathbb{P}$ represents the state transition probability function, defining the likelihood $\mathbb{P}(s'|s, a)$ of evolving to state s' from s after executing action a. Additionally, Ω denotes the observation probability function, characterizing the probability $\Omega(o|s', a)$

of perceiving observation o given the new state s'. Finally, $\gamma \in [0, 1)$ is the discount factor, and μ_0 is the initial state distribution.

In our scheduling model, we define the components of the POMDP tuple as follows:

1) State: The system state at time t, denoted as s_t, represents the complete physical environment. It comprises the request arrival vector $\boldsymbol{A}(t)$, the queue state matrix $\boldsymbol{Q}(t)$, the qubit budgets $B(t)$, the channel condition $\boldsymbol{c}(t)$, along with other relevant system parameters.

2) Observation: The observation o_t represents the local information available to the scheduler. While the queue states $\boldsymbol{Q}(t)$ and remaining budget $B(t)$ are readily observable, the instantaneous channel state $\boldsymbol{c}(t)$ is often inaccessible in practice due to measurement delays. Therefore, we formulate a flexible observation model: o_t includes $\boldsymbol{c}(t)$ in the ideal Full Observation setting while excluding it in the more realistic Partial Observation setting, forcing the agent to infer channel dynamics from historical interactions.

3) Action: Based on the observation o_t, the scheduler executes an action a_t, defined as the concatenation of qubit allocation vectors for all users: $a_t = [\boldsymbol{\nu}_1(t), \boldsymbol{\nu}_2(t), \ldots, \boldsymbol{\nu}_N(t)]$.

4) Rewards: The rewards are defined by the instantaneous weighted throughput: $r_t = U(t)$.

5) Objective: The objective is to find the policy $\pi_\theta := \pi(\cdot; \boldsymbol{\theta})$, parameterized by θ, which satisfies the constraints and maps observations to actions while maximizing the expected cumulative reward:

$$\mathcal{J}^{\pi_\theta} = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t \middle| \mu_0, \pi_\theta\right]. \tag{6}$$

Remark: Through the proposed POMDP formulation, the objective is to optimize the long-term time-average weighted throughput. Specifically, in the finite horizon setting with $\gamma \to 1$, the cumulative reward is equivalent to the total system throughput (i.e., $\sum_{t=1}^{T} r_t = T\overline{U}$). Consequently, maximizing the expected cumulative reward $\mathcal{J}^{\pi_\theta}$ implicitly solves the original scheduling problem. However, standard DRL algorithms lack intrinsic mechanisms to enforce strict resource causality, frequently resulting in invalid actions that exceed the qubit budget. To address this, we propose a novel framework that confines the policy search strictly within the feasible action space, guaranteeing that every generated schedule is physically realizable.

IV. Method

In this section, we introduce our proposed DRL algorithm, the **D**eep **Q**uantum **R**equest **S**cheduler (DQRS). It primarily consists of three components: First, a spatio-temporal branching architecture is employed to extract inter-user correlations at the current time step and the system's long-range temporal dependencies. Second, a virtual action slot is introduced to ensure decision outputs satisfy resource causality constraints. Finally, an actor-critic paradigm is utilized to achieve effective and stable policy learning within continuous action spaces. The full algorithm is presented in Algorithm 1.

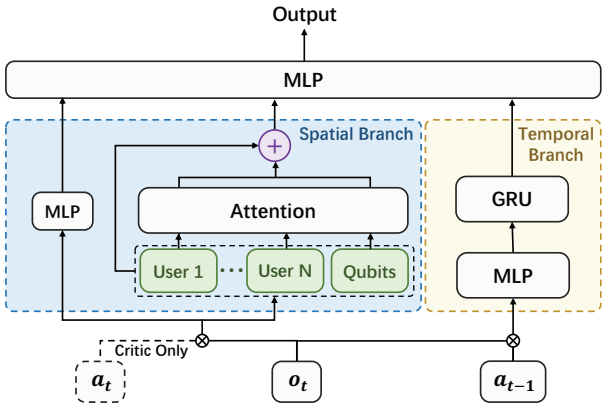


Fig. 1. The Spatio-Temporal Branching architecture of DQRS.

A. Spatio-Temporal Branching

Inspired by the temporal branch introduced in the prior work [14], we design a sophisticated spatio-temporal branching architecture for DQRS. As illustrated in Figure 1, the temporal branch employs a Recurrent Neural Network (specifically, a GRU [41] in our implementation). It utilizes the current observation and the previous action as inputs, leveraging hidden units to extract long-term temporal features to address the challenges posed by partial observability.

In contrast, the spatial branch focuses on capturing instantaneous inter-user relationships. Specifically, we treat user information (e.g., current observation and action) and available qubits as tokens and apply an attention mechanism [42] to derive correlation information. Further, to prevent the loss of original feature details during this process, we introduce a skip connection where the linearly transformed input is concatenated with the attention module's output. This design allows the network to leverage both local features and global dependencies effectively.

Finally, DQRS aggregates the results from both branches to generate the final output. With this architecture, DQRS is capable of not only handling long-range partial observability but also accurately interpreting current observational features, even in scenarios with a large number of users.

B. Virtual Action Slot

Direct optimization using standard DRL algorithms often struggles to guarantee strict adherence to resource causality constraints. To address this, we design a specialized policy structure for DQRS to ensure the generation of feasible solutions at any stage of the optimization process.

Specifically, let the dimension of the action space be denoted as $\text{Dim}_{\mathcal{A}}$. We configure the policy network to output $\text{Dim}_{\mathcal{A}}+1$ logits. These logits are normalized via a softmax function to form a probability distribution that partitions the currently available qubits (Line 7 in Algorithm 1). In this distribution, the first $\text{Dim}_{\mathcal{A}}$ dimensions correspond to the qubits actually allocated to the respective queue slots. The final dimension

serves as an additionally introduced virtual slot, which can be interpreted as the proportion of qubits reserved (i.e., not allocated) at the current time step. Furthermore, for the qubits actually allocated to each request, we apply a clipping operation to the range $[0, \nu_{max}]$ (Line 8 in Algorithm 1). This mechanism ensures that the final output actions strictly satisfy the hard constraints imposed by the available qubit resources.

C. Actor-Critic Policy Optimization

Given that the action space of our formulated scheduling problem is continuous, we employ the Actor-Critic architecture for policy optimization.

First, it is important to note that the policy structure designed in Section IV-B incorporates a non-differentiable clipping operation to enforce hard constraints. Consequently, the action optimization in DQRS is performed entirely within the logits space. However, for notational convenience, we abuse notation in the following sections to denote the actual output logits of the policy network as a, unless otherwise specified.

Subsequently, similar to prior works, to mitigate the estimation error of Q-values, we sample a batch of actions from a reference distribution μ (e.g., a uniform distribution). We then introduce a softmax operator to compute the target Q-value, which is defined as follows:

$$\text{Softmax}(Q;\beta) = \frac{\mathbb{E}_{a\sim\mu}\left[\frac{\exp(\beta Q(o,a))Q(o,a)}{\mu(a)}\right]}{\mathbb{E}_{a\sim\mu}\left[\frac{\exp(\beta Q(o,a))}{\mu(a)}\right]} \tag{7}$$

Accordingly, we optimize the critic network by minimizing the Temporal Difference (TD) error:

$$\mathcal{L}_{\text{critic}} = \left[Q(o,a) - r - \gamma \cdot \text{Softmax}\left(\bar{Q}(o',a');\beta\right)\right]^2 \tag{8}$$

where $\bar{Q}$ is the raw target Q-value, o' and a' are the next observation and action, respectively.

Conversely, the actor network is optimized to find the best policy π_θ by minimizing the negative Q-value:

$$\mathcal{L}_{\text{actor}} = -Q(o, \pi_\theta(o)) \tag{9}$$

where θ represents the trainable parameters of the policy.

In addition, we incorporate the Clipped Double Q-Learning technique, which involves maintaining two parallel actor-critic architectures (Lines 11-23 in Algorithm 1). During the training phase, we compute the target using the minimum Q-value of the two networks to suppress overestimation bias. In contrast, during deployment, the network yielding the larger Q-value is utilized for action execution. Additionally, inspired by [33], we implement a delayed policy update strategy (Lines 17-22 in Algorithm 1), where the actor is updated at a lower frequency than the critic. These mechanisms collectively ensure robust policy evaluation and enhance the overall performance of DQRS.

Algorithm 1 Deep Quantum Request Scheduler (DQRS)

1: **Input:** soft update rate κ, episode number M, episode length T, policy update frequency K.
2: **Initialize:** The replay buffer $\mathcal{D}$. The parameters φ_1, φ_2 for the critics Q_1, Q_2, and θ_1, θ_2 for the actors π_1 π_2. The parameters $\bar{\varphi}_1$,$\bar{\varphi}_2$,$\bar{\theta}_1$,$\bar{\theta}_2$ for the target networks.
3: **for** $m = 1$ **to** M **do**
4: **for** $t = 1$ **to** T **do**
5: // Interaction with the environment
6: Obtain the current observation o_t and calculate action logits $a_t = \arg\max_{j=1,2}(Q_j(o_t, a_t))$.
7: Compute the allocation ratios $\omega = \text{Softmax}(a)$, where the last ratio ω_{-1} is the virtual slot.
8: Allocate qubits $\nu_i^j = \min\left(\omega_{-1}B(t)\cdot\frac{\omega_i^j}{Q_i^j(t)},\ \nu_{\max}\right)$.
9: Store the transition $(a_{t-1}, o_t, a_t, r_t, o_t')$ into $\mathcal{D}$.
10: // Policy Update
10: Sample a batch of trajectories $\{(o_0, a_0, r_0, o_1, a_1, \ldots, o_{T+1})\}$ from $\mathcal{D}$.
11: // Double Learning
11: **for** $i = 1, 2$ **do**
12: Sample the next action logits $\hat{a}$ from π_θ^i with a Gaussian noise.
13: $\bar{Q} = \min_{j=1,2}\left(\bar{Q}_j(o, \hat{a})\right)$
14: // Softmax Operator
14: Calculate Softmax $\left(\bar{Q}(o,\hat{a});\beta\right)$ according to (7).
15: Calculate $\mathcal{L}_{critic}$ according to (8).
16: Update φ_i with $\nabla_{\varphi_i}\mathcal{L}_{critic}$.
17: // Delay Policy Update
17: **if** $t \equiv 0 \pmod K$ **then**
18: Calculate $\mathcal{L}_{actor}$ according to (9).
19: Update θ_i with $\nabla_{\theta_i}\mathcal{L}_{actor}$.
20: $\bar{\varphi}_i = \kappa\varphi_i + (1-\kappa)\bar{\varphi}_i$.
21: $\bar{\theta}_i = \kappa\theta_i + (1-\kappa)\bar{\theta}_i$.
22: **end if**
23: **end for**
24: **end for**
25: **end for**

V. Experiments

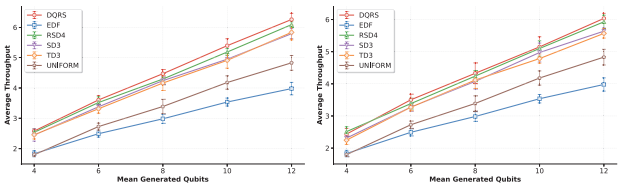
In this section, we establish a simulation environment to model the scheduling of entanglement requests within a discrete-time quantum network architecture with a star topology. Based on this framework, we evaluate the performance of DQRS. Then, we conduct an ablation study to investigate the specific contributions of its individual components. Finally, we provide visualization results to illustrate the intelligent qubits usage patterns of DQRS.

A. Setup

1) Environment: Our experimental environment consists of a scheduler serving 8 users. Request arrivals for each user follow an independent Poisson process with user-specific rates, while the global qubit generation follows an i.i.d. Exponential

TABLE I
USER CONFIGURATION.

User	Arrival Rate	Delay τ	Weight w	Fiber Length L (km)
1	5	1	5	49.68
2	5	2	4	38.31
3	2	4	2	42.63
4	4	6	1	40.05
5	5	4	2	28.67
6	5	2	1	27.47
7	4	2	5	26.84
8	5	4	3	46.26



(a) Full Observation. (b) Partial Observation.

Fig. 2. Average throughput of different algorithms with various means of generated qubits.

distribution. Any request that is not successfully served within its specific delay constraint is immediately discarded. Our experiments consider two observation settings: full observation and partial observation. In the partial observation setting, the current channel state $\boldsymbol{c}(t)$ is unobservable. Consequently, the scheduler must infer the underlying channel dynamics through interaction. The time horizon of a single episode is set to 50 time slots. For the evaluation, we execute each algorithm for 20 episodes and report the mean of the average weighted throughput. To ensure statistical reliability, all experiments are repeated with 5 different random seeds.

2) Users and Service Configuration: The specific configurations for each user, including request arrival rates, delay constraints, throughput weights, and the fiber distances from the scheduler to the users, are listed in Table I. The success probability p_i, which accounts for generation probability, transmission loss, and detection efficiency, is given by:

$$p_i = (1 - e^{-\eta\nu}) \cdot 10^{-\alpha L/10} \cdot K(c_i), \quad \nu \in [0, \nu_{\max}]. \quad (10)$$

Here, the first term $(1 - e^{-\eta\nu})$ represents the probability of successfully generating a fresh entangled pair given the amount of qubits ν, characterized by the efficiency coefficient η. The second term $10^{-\alpha L/10}$ accounts for the photon survival probability over the optical fiber of length L with attenuation coefficient α [43]. The final term $K(c_i) \in (0, 1]$ aggregates the effects of channel fading and detector efficiency corresponding to condition c_i. In our experiments, we set the model parameters as $\eta = 2.0$ and $\alpha = 0.2$ dB/km. The channel state evolution is governed by a pre-defined Markov chain that remains unknown to the scheduling algorithm. Specifically, we consider three distinct channel states, associated with coefficients $K(c)$ values of 0.1, 0.5, and 1.0, respectively.

3) Baselines: Our baselines include two representative traditional methods: ***Uniform*** and ***Earliest Deadline First (EDF)***. The former distributes available qubits equally among all requests at each time slot, while the latter prioritizes the most urgent requests. In addition, to evaluate the performance of our proposed algorithm architecture, we benchmark against three advanced DRL algorithms: ***TD3*** [33], ***SD3*** [34], and ***RSD4*** [14]. It is worth noting that for all DRL algorithms, we incorporate the Virtual Action Slot mechanism to guarantee the feasibility of the learned policies. For the training configuration, γ is set to 0.99, and all DRL models are trained over $5,000$ episodes.

B. Evaluation

To assess the performance of DQRS and the baselines across different qubit generation scenarios, we vary the mean of the exponential distribution over five specific values in this section. The initial qubit budget is set to $B_{\text{init}} = 10.0$.

The experimental results are presented in Figure 2. Leveraging our proposed virtual action slot mechanism, all DRL algorithms discovered policies superior to traditional methods under strict qubit resource causality constraints, thereby validating the effectiveness of this mechanism.

In detail, under the full observation scenario (Figure 2a), all DRL algorithms significantly outperform the two traditional methods. Notably, DQRS achieves superior performance among all evaluated algorithms, surpassing the best traditional baseline by up to **44%** and the best-performing DRL baseline by up to **4%**.

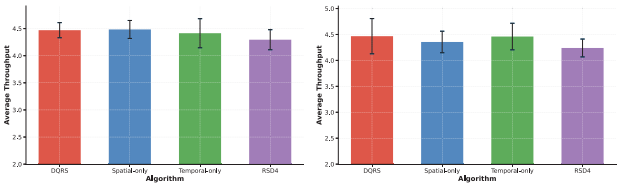
Specifically, compared to RSD4, which relies solely on a temporal branch and a full connection branch, DQRS leverages its additional spatial branch to effectively capture correlations among multiple users, thereby further enhancing its performance.

In the partial observation setting (Figure 2b), the absence of channel state information leads to a performance degradation across all DRL algorithms. However, they still maintain a substantial advantage over traditional approaches. Even in this more challenging environment, DQRS demonstrates performance comparable to, or even better than, other DRL counterparts, highlighting its robustness across diverse scenarios.

C. Ablation Study

A critical component of DQRS is its spatio-temporal branching architecture. To evaluate its impact, we conduct an ablation study under a mean generated qubits of 8.0, considering both full and partial observation scenarios. Specifically, in addition to the full DQRS model, we evaluate two variants: (1) ***Spatial-only***, which removes the GRU-based temporal branch; and (2) ***Temporal-only***, which omits the attention-based spatial branch and feeds the current action a_t directly into the temporal branch for the critic networks. We also benchmark against RSD4, the current state-of-the-art DRL baseline.

The results are illustrated in Figure 3. It is evident that the *Spatial-only* variant consistently outperforms RSD4 across



(a) Full Observation. (b) Partial Observation.

Fig. 3. Ablation on branches (Mean Generated Qubits = 8).

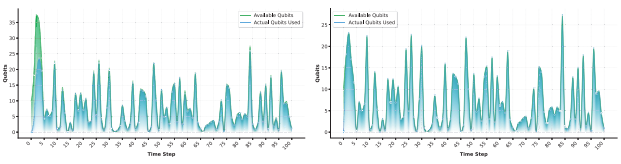
both observation scenarios. Furthermore, in the full observation setting, it achieves performance comparable to DQRS and superior to the *Temporal-only* variant. This indicates that the attention-based spatial branch effectively enhances feature extraction capabilities for complex multi-user states, contributing significantly to the superior performance of DQRS.

On the other hand, the inclusion of the temporal branch yields distinct benefits. In the full observation setting, while the temporal branch has a marginal impact on the average throughput of DQRS, it reduces the variance, thereby stabilizing performance in time-varying quantum environments. Conversely, in the partial observation setting, where instantaneous channel state information $\boldsymbol{c}(t)$ is unavailable and agents must infer channel dynamics through interaction, both DQRS and the *Temporal-only* variant outperform the *Spatial-only* variant. This suggests that the temporal branch empowers DQRS to capture these temporal dependencies, leading to further performance gains over the *Spatial-only* variant.

D. Visualization on Qubits Consumption

In the partially observable quantum network, the lack of precise system state information compels traditional methods to adopt a greedy approach, exhausting available qubits at each time step. Conversely, a key factor contributing to the superior efficiency of DRL algorithms, represented by DQRS, is their ability to perceive system dynamics through interaction, thereby enabling more flexible qubit allocation. For instance, the agent can strategically conserve qubits during specific periods to reserve resources for future high-priority packets or superior channel conditions, ultimately enhancing transmission success rates and overall performance.

To visually illustrate these distinct qubit usage patterns, Fig. 4 depicts the available qubits $B(t)$ (green curve) and actual qubit consumption (blue curve) for both DQRS and the Uniform algorithm. The data is recorded over a 100-step episode under the full observation setting with a mean generated qubits of 8.0. For this visualization, DQRS utilizes a policy model trained on a 50-step horizon (i.e., the model trained in Section V-B), with data points connected via interpolation for smoothness. It is evident that while the Uniform algorithm exhausts all available qubits at every time step, DQRS demonstrates intelligent qubit management. For example, it conserves qubits during initial phases of low request traffic to accommodate subsequent high-load stages, thereby highlighting the superiority of the DQRS algorithm.



(a) DQRS. (b) Uniform.

Fig. 4. Available Qubits v.s. Actual Qubits Used in an episode (Full Observation, $T = 100$, Mean Generated Qubits = 8).

VI. Conclusion

In this paper, we addressed the critical challenge of entanglement request scheduling in quantum networks, explicitly accounting for strict delay constraints, partial observability, and time-varying resource budgets. By formulating the problem as a Partially Observable Markov Decision Process, we proposed the Deep Quantum Request Scheduler (DQRS). To tackle the system's complexity, DQRS integrates a spatio-temporal branching architecture for effective feature extraction within an Actor-Critic framework, and introduces a novel virtual action slot mechanism to strictly enforce resource causality in the continuous action space. Extensive simulation results demonstrate that, under various resource budgets and observation settings, DQRS achieves performance gains of up to **44%** over traditional heuristics and **4%** over state-of-the-art DRL baselines, underscoring its robustness and effectiveness in resource-constrained quantum networks.

References

[1] L. Le and T. N. Nguyen, "Dqra: Deep quantum routing agent for entanglement routing in quantum networks," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–12, 2022.

[2] Y. Gao, S. Yang, F. Li, Y. Li, L. Zhu, S. Trajanovski, and X. Fu, "Efficient and flexible multi-qubit entanglement transmission in quantum networks," *IEEE Transactions on Networking*, vol. 34, pp. 1761–1776, 2026.

[3] H. Gu, Z. Li, X. Wang, D. Yang, G. Xue, and R. Yu, "Cost-aware high-fidelity entanglement distribution and purification in the quantum internet," *IEEE Transactions on Networking*, vol. 34, pp. 681–696, 2026.

[4] P. Fittipaldi, A. Giovanidis, and F. Grosshans, "A linear algebraic framework for dynamic scheduling over memory-equipped quantum networks," *IEEE Transactions on Quantum Engineering*, 2023.

[5] T. Vasantam and D. Towsley, "Stability analysis of a quantum network with max-weight scheduling," *arXiv preprint arXiv:2106.00831*, 2021.

[6] Y. Chen, L. Huang, and J. C. S. Lui, "Towards stability and utility maximization in adversarial quantum networks under bandit feedback (invited)," in *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, 2025, pp. 347–354.

[7] P. Promponas, V. Valls, S. Guha, and L. Tassiulas, "Maximizing entanglement rates via efficient memory management in flexible quantum switches," *IEEE Journal on Selected Areas in Communications*, 2024.

[8] H. Zhou, K. Lv, L. Huang, and X. Ma, "Quantum network: Security assessment and key management," *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1328–1339, 2022.

[9] T. Islam, M. Arifuzzaman, and E. Arslan, "Reinforcement learning based proactive entanglement swapping for quantum networks," in *2024 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, 2024, pp. 135–142.

[10] Á. T. Olivas, A. A. Casado, M. P. González, J. Faba, L. M. Robledo, V. Martin, and L. Ortiz, "Reinforcement learning for entanglement swapping in quantum networks," in *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, 2025, pp. 274–278.

[11] Y. Gan, S. Ling, R. Zhou, L. Shen, and C. Qian, "Qplan: Deep reinforcement learning assisted quantum network planning," in *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, 2025, pp. 363–370.

[12] R. Singh and P. R. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 127–142, 2019.

[13] K. Chen and L. Huang, "Timely-throughput optimal scheduling with prediction," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2457–2470, 2018.

[14] P. Hu, Y. Chen, L. Pan, Z. Fang, F. Xiao, and L. Huang, "Multi-user delay-constrained scheduling with deep recurrent reinforcement learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 3, pp. 2344–2359, 2024.

[15] Z. Li, P. Hu, and L. Huang, "Offline learning-based multi-user delay-constrained scheduling," in *2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, 2024, pp. 92–99.

[16] X. Wang, Z. Li, and L. Huang, "Beyond static populations: Efficient delay-constrained scheduling for dynamic users via deep reinforcement learning," in *Proceedings of the Twenty-Sixth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, ser. MobiHoc '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 151–160.

[17] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 490–499.

[18] L. Yang, Y. Zhao, H. Xu, and C. Qiao, "Online entanglement routing in quantum networks," in *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. IEEE, 2022, pp. 1–10.

[19] L. Yang, Y. Zhao, L. Huang, and C. Qiao, "Asynchronous entanglement provisioning and routing for distributed quantum computing," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[20] Y. Zhao and C. Qiao, "Distributed transport protocols for quantum data networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 2777–2792, 2023.

[21] Y. Gan, X. Zhang, R. Zhou, Y. Liu, and C. Qian, "A routing framework for quantum entanglements with heterogeneous duration," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2023, pp. 1132–1142.

[22] N. K. Panigrahy, P. Dhara, D. Towsley, S. Guha, and L. Tassiulas, "Optimal entanglement distribution using satellite based quantum networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2022, pp. 1–6.

[23] S. Pouryousef, N. K. Panigrahy, and D. Towsley, "A quantum overlay network for efficient entanglement distribution," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[24] G. Zhao, J. Wang, Y. Zhao, H. Xu, L. Huang, and C. Qiao, "Segmented entanglement establishment with all-optical switching in quantum networks," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 268–282, 2023.

[25] N. K. Panigrahy, M. G. De Andrade, S. Pouryousef, D. Towsley, and L. Tassiulas, "Scalable multipartite entanglement distribution in quantum networks," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2. IEEE, 2023, pp. 391–392.

[26] H. Gu, R. Yu, Z. Li, X. Wang, and F. Zhou, "Esdi: Entanglement scheduling and distribution in the quantum internet," in *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2023, pp. 1–10.

[27] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Entanglement routing design over quantum networks," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 352–367, 2023.

[28] M. Chehimi, M. Elhattab, W. Saad, G. Vardoyan, N. K. Panigrahy, C. Assi, and D. Towsley, "Reconfigurable intelligent surface (ris)-assisted entanglement distribution in fso quantum networks," *IEEE Transactions on Wireless Communications*, 2025.

[29] N. K. Panigrahy, T. Vasantam, D. Towsley, and L. Tassiulas, "On the capacity region of a quantum switch with entanglement purification," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[30] G. Vardoyan, P. Nain, S. Guha, and D. Towsley, "On the capacity region of bipartite and tripartite entanglement switching," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 8, no. 1-2, pp. 1–18, 2023.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[33] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1587–1596.

[34] L. Pan, Q. Cai, and L. Huang, "Softmax deep double deterministic policy gradients," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 11 767–11 777.

[35] P. Dong, Q. Li, D. Sadigh, and C. Finn, "Expo: Stable reinforcement learning with expressive policies," *arXiv preprint arXiv:2507.07986*, 2025.

[36] H. Zhong, X. Wang, Z. Li, and L. Huang, "Offline-to-online multi-agent reinforcement learning with offline value function memory and sequential exploration," in *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 2025, pp. 2373–2381.

[37] X. Wang, Z. Li, H. Zhong, and L. Huang, "Beyond shallow behavior: Task-efficient value-based multi-task offline marl via skill discovery," *arXiv preprint arXiv:2502.08985*, 2025.

[38] C. Simon, H. De Riedmatten, M. Afzelius, N. Sangouard, H. Zbinden, and N. Gisin, "Quantum repeaters with photon pair sources and multimode memories," *Physical review letters*, vol. 98, no. 19, p. 190503, 2007.

[39] O. Collins, S. Jenkins, A. Kuzmich, and T. Kennedy, "Multiplexed memory-insensitive quantum repeaters," *Physical review letters*, vol. 98, no. 6, p. 060502, 2007.

[40] H. Song, M. Xiao, J. Xiao, Y. Liang, and Z. Yang, "A pomdp approach for scheduling the usage of airborne electronic countermeasures in air operations," *Aerospace Science and Technology*, vol. 48, pp. 86–93, 2016.

[41] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[43] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of modern physics*, vol. 74, no. 1, p. 145, 2002.