

Predictive Fidelity-Aware Routing and Adaptive Scheduling in Distributed NISQ Quantum Networks via Actor–Critic Reinforcement Learning

Prabu Pachiyannan

ppachiyannan@imamu.edu.sa

Imam Mohammad ibn Saud Islamic University

Article

Keywords: Quantum communication, Deep reinforcement learning, Fidelity prediction, Adaptive scheduling, Distributed NISQ systems

Posted Date: July 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9250810/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Predictive Fidelity-Aware Routing and Adaptive Scheduling in Distributed NISQ Quantum Networks via Actor–Critic Reinforcement Learning

¹Prabu Pachiyannan

¹Prabu P, Information Systems Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia

Abstract

Quantum communication plays a key role in enabling distributed Noisy Intermediate-Scale Quantum (NISQ) systems; however, practical quantum networks are often affected by link fluctuations, limited quantum resources, and entanglement degradation. These factors make reliable routing and scheduling difficult, particularly in large-scale and dynamically changing environments. Classical methods such as Historical Weighted Fidelity Routing (HWFR) and Dynamic Multi-Priority Quantum Scheduling (DMPQS) generally perform well only when network conditions remain stable, and their performance degrades when entanglement success rates vary or relay nodes become congested. A major challenge therefore lies in operating under uncertain and time-varying fidelity conditions. Most existing approaches focus on either routing or scheduling in isolation and do not explicitly consider future fidelity variations, which can lead to inefficient resource usage and increased end-to-end latency. In this work, we propose a Deep Reinforcement Learning-based Quantum Communication Optimization Framework (DRL-QCOF) that jointly addresses routing and scheduling. The framework employs a Bi-LSTM-based Quantum Fidelity Prediction Network (QFPN) to estimate future link fidelity $F_{\text{pred}}(t)$ from time-series features including link length, noise coefficient, entanglement success probability, and relay utilization. These predictions are then used by a Deep Reinforcement Learning-based Adaptive Quantum Scheduling Agent (DRAQS-Agent), implemented using an Actor–Critic architecture with a continuous reward formulation, to guide routing and scheduling decisions across distributed nodes. Simulation results indicate that the proposed approach improves the average link fidelity to a maximum of 0.88, compared to 0.78–0.86 achieved by existing methods. The end-to-end latency is reduced from 104 ms to 83 ms, while baseline schemes remain above 110 ms under the same conditions. In addition, the average response time is reduced to approximately 147 ms, resource utilization increases to 71–83%, and the success ratio improves from 93% to 97%, resulting in improved end-to-end reliability. The cumulative reward stabilizes at around 16.0, with convergence observed within 300 training episodes, and the achieved throughput reaches 118–121 entangled pairs per second, confirming the effectiveness of the proposed framework in dynamic NISQ network scenarios.

Keywords:

Quantum communication, Deep reinforcement learning, Fidelity prediction, Adaptive scheduling, Distributed NISQ systems

1. Introduction

The rapid expansion of the intelligent communication infrastructures has transformed how the modern networks operate in environments that demands for reliable, low-latency and an adaptive control. Recent studies have shown that the advanced networked systems have continued to evolve towards the complex, service-driven architectures that depend on a seamless coordination among the distributed resources [1–3]. These systems include the quantum-aware communication layers, software-defined networks and an AI-enabled control protocols, which together attempt to meet the real-time service requirements [4–6]. The demand for a greater scalability has encouraged the use of a dynamic routing, predictive decision-making and an autonomous optimization model that tends to operate across the large control horizons [7–8]. In such settings, the network continuously has adjusted various parameters to sustain the high fidelity, low delay, and stable connectivity, while the underlying decision process have grown more intricate as the system scales.

Despite these advancements, several works have shown the presence of persistent challenges that have continued to limit the real-world adoption of the intelligent control frameworks [9–11]. One of the major issues involves the sensitivity of routing decisions to the fluctuating traffic loads, which causes the instability in a high-density network. The presence of heterogeneous devices further has complicated the link availability and resource allocation, which often leads to the unpredictable latency patterns. Several studies also state that learning-based strategies have struggled with the generalization, particularly when the environment has introduced an irregular variation in the packet arrival rates or queueing dynamics [12–13]. Another difficulty arises from the limitations is the convergence speed. Many of the existing algorithms requires the prolonged training cycles and it produce an inconsistent performance during the early learning stages, which has restricted the operational use. Systems also have encountered the constraints in balancing the exploration and exploitation, which results in suboptimal use of an available resources [14–15]. Collectively, these challenges have hindered the goal of maintaining a reliable performance in the fast-changing network conditions.

The core problem discussed in the recent research centers on an inability of the current control strategies to maintain a stable link fidelity, predictable delays and efficient resource utilization under the long horizon learning scenarios [16]. When the system has experienced the constant fluctuations, conventional methods often have relied on a static or semi-adaptive strategy that do not respond quickly. These limitations have leads to the high blocking ratios, increased latency and a reduced service quality, which significantly has undermined the overall performance of the mission-critical networks. Although, reinforcement learning (RL) and deep neural models (DNN) have been introduced to improve the adaptability, many of the existing solutions remain either to be computationally heavier one or sensitive to the model drift. As a result, they fail to consistently deliver a stable and high-quality routing decisions.

Most of the existing studies have explored the learning-enabled routing or dynamic control only up to moderate system complexities. Few of the works have investigated how the model performs when it is evaluated across the extended control horizons that requires the thousands of training epochs and the frequent re-optimization. The existing

literature rarely has examined how the multiple metrics have evolved together, particularly under the unpredictable traffic variations. Metrics such as link fidelity, end-to-end latency, resource utilization, convergence episodes, blocking rate and the cumulative reward often remain evaluated in an isolation. This fragmented assessment prevents a deep understanding of how the learning models behave when it jointly optimizes the interdependent network properties. Further, most of the studies do not address the difficulty of proving simultaneously a higher throughput and a lower delay in a highly dynamic environment. Therefore, the field have lacked a unified, multi-metric and the long-horizon evaluation approach that tends to show a realistic operating condition.

Considering these gaps, the objectives of this study focus on creating a robust control strategy that responds effectively to the rapid network changes while it maintains a higher performance across the diverse metrics. The work aims to design a policy that learns a stable behavior within the larger control epochs and it maintains a consistent performance during the training and deployment. Another major objective is to improve the resource utilization by the minimization of blocking, lowering latency and the increased throughput. The framework also seeks to reduce the training instability by proving that the model converges within fewer episodes. In addition, the study aims to analyze how the proposed method simultaneously has improved the fidelity, latency, throughput, cumulative reward and the convergence behavior. These objectives have proved a practical direction towards the construction of a reliable, efficient and an adaptable network control solution.

The novelty of this work arises from its unified perspective on a multi-metric optimization across the extended training horizons. Rather than prioritizing a single performance metric, the proposed approach has combined several interdependent network attributes into a consolidated decision structure that has guided the learning process. This design has allowed the model to understand the trade-off, and it adjusts dynamically with the evolving conditions. Another novel aspect involves an incorporation of stability-driven reward shaping, which has encouraged the agent to prioritize a long-term reliability rather than the short-term improvements. This mechanism has helped the model to avoid fluctuations that typically appear during the training. The study also has introduced an adaptive state representation that tends to capture both the transient and the cumulative network dynamics. This hybrid design provides a more of an accurate context for decision-making, which also reduces the chances of unstable or inefficient routing. The exploration strategy has also been refined to balance performance and an adaptability, allowing the agent to adjust more of an effectively during the sudden changes in traffic load.

Based on these elements, the major contributions of this study are summarized as follows.

- 1) First, the work proposes a RL-based control strategy that optimizes the routing and the resource decisions across the long training horizons, which proving a consistent performance improvement as the system evolves.
- 2) Second, the model has incorporated a multi-metric reward structure that has aligned the learning process with the practical operational goals, which includes

an improved link fidelity, reduced latency, lower blocking rate and the higher throughput.

- 3) Third, the study has designed an adaptive state formulation that tends to capture a richer representation of the network dynamics, which has supported for more of an accurate and a stable decision patterns.

2. Related Works

Research on quantum network routing has continued to expand the as quantum communication, QKD systems and the entanglement distribution have evolved towards the practical deployment.

Secure Routing and Resource Constraints in QKD Networks

QKD-based routing sets the foundation for understanding how the limited quantum keys, physical-layer variability, and the strict security requirements have influenced the network performance. Akhtar et al. [17] have investigated the fundamental issue of the fast and secure packet delivery in the multi-hop QKD networks. It then considers a trusted-node setup in which the symmetric key pairs are generated, which are used to encrypt the packets before transmission. The main challenge arises here is due to time-varying key availability and the heterogeneous link capacities. To solve this, it proposes the Tandem Queue Decomposition (TQD) policy, which builds upon the Universal Max Weight scheme. The TQD approach has transformed the original routing problem into a generalized network flow formulation that has removed the explicit key constraints. Its Lyapunov-based analysis shows that the policy has achieved a full secure capacity region for diverse traffic patterns.

Chen et al. [18] extend the discussion by the use of a demand-aware routing framework for QKDNs, termed ADA-QKDN. It designs a routing management center using the software-defined networking principles, which has reduced the local computational burden on quantum nodes. Its policy treats the application demands for as a key factor, which allows the dynamic adjustment of link evaluation indicators. The simulation results have shown that the method has increased request success rate by more of a than 10%, while it maintains an effective key rate comparable to the shortest-path-based schemes.

Wang et al. [19] has increased the scope by presenting a general mathematical framework for quantum routing using the routing algebra to model the probabilistic behaviour of quantum links. Its algorithms are provably optimal and an has addressed fairness among the simultaneous routing requests.

Entanglement Routing, Fidelity Assurance, and the Multi-Path Management

Once QKD-centric routing matures, another line of studies focuses on entanglement distribution, fidelity control, and the multi-path routing in quantum repeater networks. Li et al. [20] has addressed the problem of entanglement generation across the lattice networks that tends to operate under the finite capacities. Its routing approach has allowed multiple paths per request while it maintains a fidelity through purification. It develops several algorithms for the capacity allocation, which includes the propagatory update

method, which has used the proportional share concepts and a progressive filling. The evaluation has shown the improvements in scheduling performance.

Cicconetti et al. [21] has examined the resource allocation for distributed computing tasks from a quantum network provisioning standpoint. The simulations reveal performance constraints, informing future directions for combining the smart computing requirements into the quantum routing.

Sharma et al. [22] has combined deep RL into the QKD optical networks. It focusses on routing and the resource assignment in the quantum signal channel. Its DRL-based scheme learns route and the resource decisions using the neural networks. Performance evaluation across the NSFNET and the UBN24 has shown blocking reduction up to 33.50% and the improvements in resource utilization up to 7.18%, compared with the DQN, FF, and the RF schemes. This marks an early shift from static or algebraic approaches towards the learning-driven and the environment-aware quantum routing.

Khudair Madhloom et al. [23] contribute a quantum-inspired swarm intelligence method that has improved gateway discovery in wireless contexts. Its QACO approach uses quantum parallelism and the entanglement concepts to have explored the routing paths more of an efficiently. Its experimental results show major improvements, which includes 70% the faster discovery of gateways and the large reductions in routing and the gateway overhead. This work connects classical networking problems with the quantum-inspired optimization, bridging the field towards the hybrid designs.

Li et al. [24] offer a purification-enabled routing framework designed to guarantee fidelity for both the single and the multiple S-D pairs. Its Q-PATH algorithm optimizes purification decisions, while it Q-LEAP has reduced computational cost by extending Dijkstra's algorithm. A greedy approach manages multiple S-D requests. Its results have shown better throughput and the higher end-to-end fidelity, which is essential for the scalable quantum repeater networks.

Yang et al. [25] have explored the asynchronous routing protocols for the quantum networks. It use dynamic topologies that the allow nodes to update the entanglement-link states asynchronously. Its method identifies optimal swapping paths while it preserves the unused direct entanglements. Its results show that the asynchronous strategies achieve higher entanglement rates than synchronous approaches, and the performance has increased with the coherence time.

Quantum-Inspired Optimization for Routing Problems

While it the previous studies have focussed directly on the quantum network routing, another segment of the literature has explored the quantum-assisted optimization techniques that indirectly supports the routing research. These works show how the quantum-inspired computing may help to solve the classical routing problems that share the structural similarities with the quantum communication systems.

Fitzek et al. [26] has evaluated the quantum approximate optimization algorithm (QAOA) for the heterogeneous vehicle routing problem (HVRP). It then maps the routing problem onto a Hamiltonian and it simulate up to 21 qubits. The results reveal that qubit

requirements scale quadratically with the number of customers. The comparison of classical optimizers in QAOA exposes a trade-off between the solution quality and the runtime.

Urgelles et al. [27] has applied QAOA to routing in wireless mesh networks. This study has shown both the single-objective and the multi-objective routing formulations, and it discusses the potential onset of quantum supremacy for these problems.

Chen et al. [28] has designed a hybrid greedy and the dynamic programming algorithm for entanglement generation. It models the swapping process using an entanglement graph that is NP-complete. To manage complexity, it breaks the graph into the sub-problems that the DP solver addresses efficiently. The simulation results show a higher entanglement distribution rates. The DP-based decomposition contributes a polynomial-time alternative to fully quantum or heuristic-based routing.

Osaba et al. [29] survey quantum optimization techniques in depth, identifying common study types, hybrid strategies, and the frequently used quantum hardware. The overview has contextualized the rapid expansion of quantum routing and the quantum optimization research.

Dixit et al. [30] has formulated the stochastic time-dependent shortest path problem as a quadratic constrained binary optimization problem. It shows that the problem has grown exponentially with an independent link cost. It then shows that the quantum solver produces a linear computational behaviour as the problem size increases.

Dornemann et al. [31] has proposed a supervised deep learning method for the capacitated VRP with time windows. The approach uses a graph convolutional network to guide the tree search. It also presents a QUBO-based formulation that is solved using the quantum-inspired computing. Santos et al. [32] studied the quantum networking using routing via a purification-aware shortest paths. It designs a multi-objective routing metric that guarantees end-to-end fidelity and the discards inferior paths.

Table 1: Summary of Methods

Method (Ref)	Algorithm / Model	Methodology	Outcomes
Akhtar et al. [17]	Tandem Queue Decomposition (TQD)	Integrates QKD key-generation dynamics into Universal Max Weight routing and transforms the secure routing problem into a generalized network-flow model	Achieves secure capacity region; supports unicast, broadcast, multicast, anycast; improves performance on OMNeT++ simulations

Chen et al. [18]	ADA-QKDN Routing	Uses SDN-based routing center, adapts paths based on application demand, adjusts link metrics dynamically	Increases request success rate by >10%; maintains effective key rate similar to SPF-based schemes
Wang et al. [19]	Routing Algebra + Near-optimal Heuristics	Analyzes quantum metrics through routing algebra; proposes fairness-aware optimization under quantum constraints	Provides provably optimal routing and scalable near-optimal solutions for complex quantum traffic
Li et al. [20]	Proportional Share, Progressive Filling, Propagatory Update	Uses multi-path routing with edge-capacity scheduling; has allowed fidelity via purification per path	Propagatory update improves scheduling; enhances throughput and resource allocation for entanglement requests
Cicconetti et al. [21]	Quantum-aware Resource Allocation	Evaluates resource provisioning for distributed smart computing nodes via simulation	Identifies bottlenecks; highlights challenges for scalable quantum resource management
Sharma et al. [22]	DRL-RRA	Uses deep reinforcement learning to assign routes and resources for QKD optical networks	Reduces blocking up to 33.5%; improves resource utilization up to 7.18%
Khudair Madhloom et al. [23]	Quantum-Inspired ACO (QACO)	Applies quantum swarm principles for gateway discovery and dynamic route repair	Improves GW discovery time (27–70%); reduces routing and GW overhead (53–60%)
Li et al. [24]	Q-PATH, Q-LEAP	Designs purification-aware routing for fidelity guarantees; uses extended Dijkstra for low complexity	Improves throughput, resource utilization, and end-to-end fidelity
Yang et al. [25]	Asynchronous Routing Protocols	Maintains dynamic topology, updates	Achieves higher entanglement rate and scalability under

		entanglement links asynchronously	longer coherence times
Fitzek et al. [26]	QAOA for HVRP	Maps routing to Ising Hamiltonian, tests QAOA with classical optimizers	Shows scalable qubit requirements; identifies performance–runtime trade-offs
Urgelles et al. [27]	QAOA-based Mesh Routing	Uses single- and multi-objective QAOA formulations	Demonstrates feasibility of quantum machine learning for routing optimization
Chen et al. [28]	Greedy + DP-based Swapping	Divides EG into solvable sub-problems using dynamic programming	Achieves high entanglement distribution rate with polynomial runtime
Osaba et al. [29]	Survey + Device Analysis	Reviews quantum routing / optimization research landscape	Summarizes trends, gaps, hardware preferences
Dixit et al. [30]	QUBO-based STDSP	Formulates routing as quadratic constrained binary optimization	Shows quantum solvers scale linearly relative to problem size
Dornemann et al. [31]	GCN-assisted QUBO	Uses neural guidance and quantum-inspired optimization	Outperforms classical solvers on large CVRPTW instances
Santos et al. [32]	Multi-objective Quantum Routing	Uses fidelity aware, path-extension algorithm with purification	Achieves near-linear complexity; has allowed fidelity thresholds and optimal path selection

Quantum communication networks have continued to have evolved as foundational elements for the distributed quantum computing. However, the current literature has revealed several unresolved issues that limit the performance, adaptability, and the reliability of these networks in realistic NISQ-era environments. Existing routing and the scheduling schemes have relied heavily on static or heuristic-based decision rules that struggle when conditions fluctuate rapidly. Methods such as a HWFR or DMPQS perform reasonably in a stable network setting, however it fail to respond effectively when link fidelity varies due to noise bursts, decoherence, or congestion at quantum repeaters. These methods are not built to anticipate changes before the It affect entanglement quality, which have led to an increased latency, inefficient resource usage, and the degraded communication success rates.

Most of the existing approaches either have focussed solely on routing or has addressed scheduling as a separate optimization problem. This separation creates a structural limitation because routing decisions have influenced the scheduling efficiency, and the vice versa, particularly when entanglement success probabilities shift within short timescales. The surveyed literature also shows that Many of the solutions offer deterministic or rule-based strategies, which cannot capture the nonlinear interactions between the entanglement fidelity, path length, quantum memory availability, and the relay load. Machine learning-based techniques appear in the recent work, but it often lacks predictive capability for the fidelity evolution or it does not combine learning with the simultaneous routing–scheduling decisions.

Thus, there is a need for a combined framework that tends to predict the link fidelity proactively and the uses those predictions to guide the real-time routing and the scheduling. A learning-based architecture that adapts to the stochastic quantum environments to improve the network performance and to support a scalable distributed quantum computing.

3. Proposed DRL-QCOF

The proposed DRL-QCOF operates as an intelligent control framework that uses predictive learning and an adaptive RL to optimize the routing and the scheduling jointly. The method has two interconnected components:

1. **Quantum Fidelity Prediction Network (QFPN):** The QFPN has received the sequences of historical link parameters, and it estimates the near-future fidelity. The Bi-LSTM captures both the forward and the backward temporal correlations, which allows the model to recognize fluctuations that affects the entanglement quality. The predicted fidelity acts as the primary performance indicator for route evaluation.
2. **Deep Reinforcement-based Adaptive Quantum Scheduling Agent (DRAQS-Agent):** The DRAQS-Agent has interacted continuously with the environment. It uses an Actor–Critic model, in which the Actor chooses the routing and the scheduling actions, and the Critic evaluates the long-term value of those actions.

The agent learns a reward that maximizes fidelity while it reduces the latency, congestion, and the overload. Together, the QFPN and the DRAQS-Agent produce an adaptive pipeline that tends to have predicted the quantum link state, selects the optimal path, and it has adjusted scheduling under the dynamic NISQ conditions.

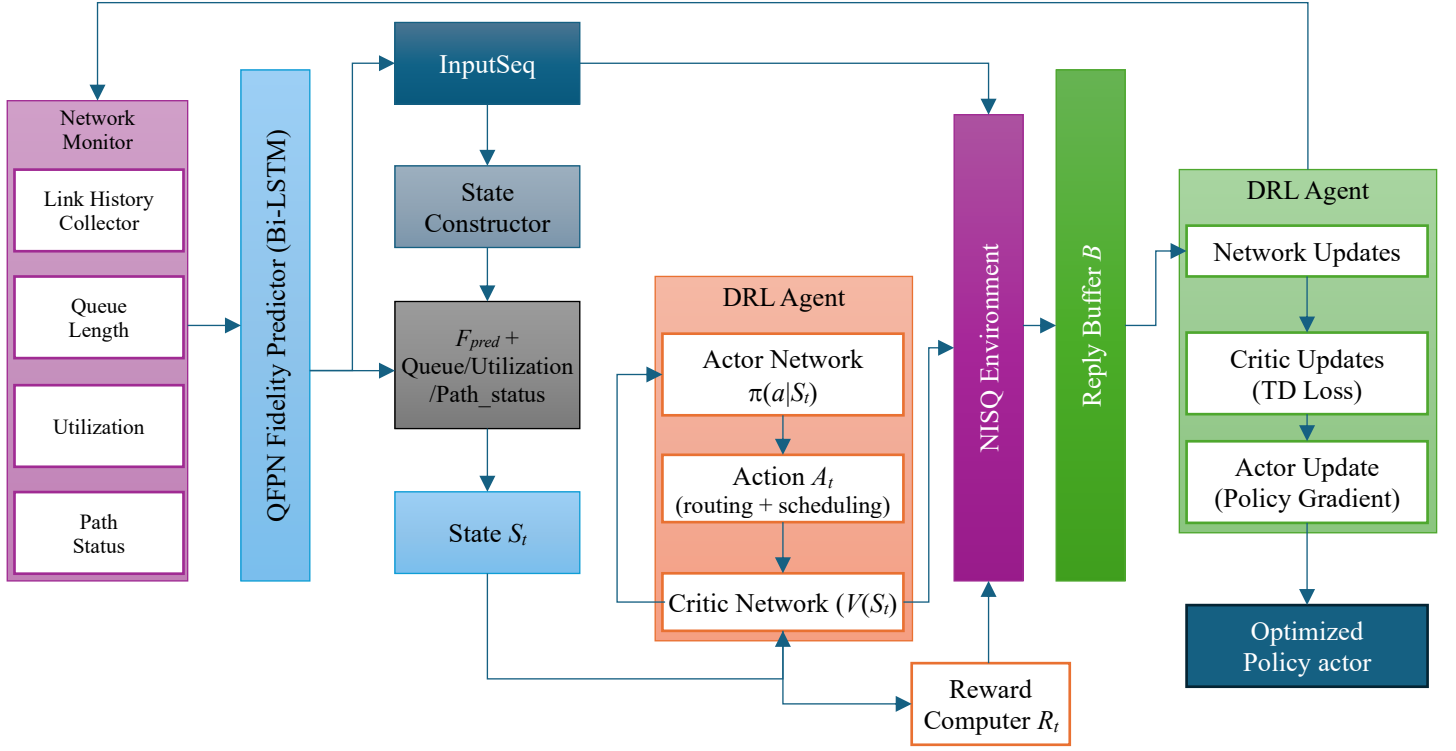


Figure 1: Foundational Diagram of DRL-QCOF

Pseudocode for DRL-QCOF

```

Initialize QFPN (Bi-LSTM), Actor network, Critic network
Initialize replay buffer B
for episode = 1 to MaxEpisodes do
  Reset environment
  for t = 1 to T do
    # Step 1: Fidelity Prediction
    InputSeq = Collect_Link_History()
    F_pred = QFPN.predict(InputSeq)
    # Step 2: Construct State
    State_t = [F_pred, queue_length, utilization, path_status]
    # Step 3: RL Action
    Action_t = Actor(State_t)
    # Step 4: Execute Action
    Env_Response = Apply_Routing_Scheduling(Action_t)
    # Step 5: Observe Reward
    Reward_t = ComputeReward(fidelity, latency, overload)
    # Step 6: Store into the Replay Buffer
    B.add(State_t, Action_t, Reward_t, State_t+1)
    # Step 7: Update Networks
    Critic.update(B)
    Actor.update(B, Critic)
  end for
end for
Return Optimized Routing and the Scheduling Policy

```

3.1. Quantum Fidelity Prediction Network (QFPN)

The QFPN as in figure 2 serves as the predictive backbone of the DRL-QCOF framework. Its purpose lies in estimating the time-varying link fidelity across the distributed NISQ networks, where each quantum channel has experienced the continuous fluctuations caused by noise, decoherence, and the entanglement failures. The QFPN uses a Bi-directional Long Short-Term Memory (Bi-LSTM) architecture that tends to capture sequential dependencies in both the forward and the backward temporal directions, which allows the model to learn nuanced patterns in historical link behavior. This predictive capability has guided the DRAQS-Agent towards the optimal routing and the scheduling decisions. The following sections describe an internal data structure, mathematical workflow, and the operational mechanics of the QFPN with the carefully constructed equations and the tables.

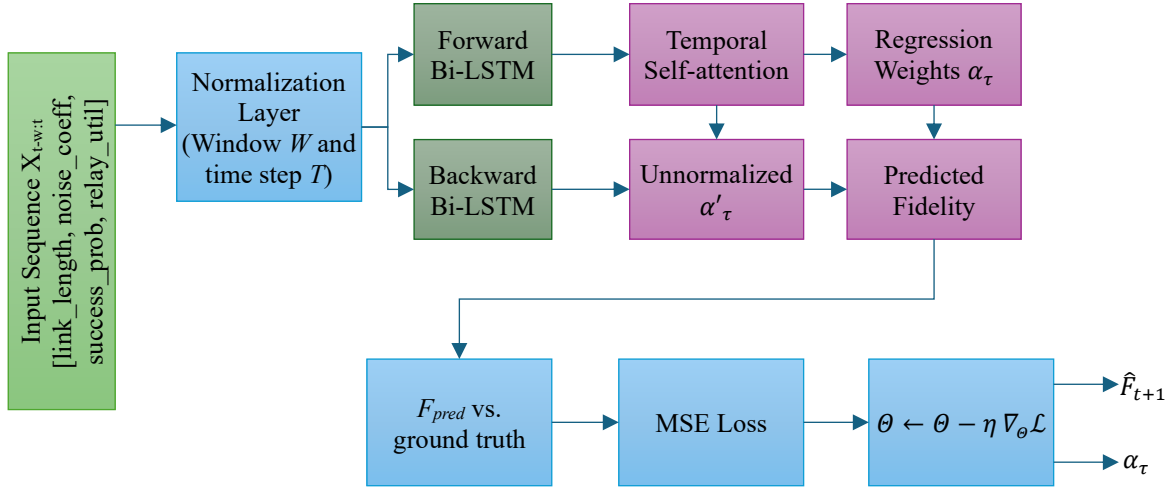


Figure 2: Architecture of QFPN

Quantum Fidelity Prediction Network (QFPN)

To predict near-future link fidelity $F_{\text{pred}}(t + 1)$ from a short history of link features so the scheduler may act proactively.

Input

- A windowed sequence of raw features for a given link:
- $\mathcal{X}_t = \{\mathbf{x}_{t-k+1}, \mathbf{x}_{t-k+2}, \dots, \mathbf{x}_t\}$, where each $\mathbf{x}_\tau = [l_\tau, n_\tau, s_\tau, u_\tau]^\top$ denotes link length l , noise coefficient n , entanglement success probability s , and relay utilization u .
- Trained QFPN parameters: Bi-LSTM weights Θ_{LSTM} , attention weights W_a, v , regression weights W_r, b_r .
- Normalization statistics μ, σ for each feature.

Output: Scalar predicted fidelity $F_{\text{pred}}(t + 1) \in [0, 1]$, Per-time-step attention weights $\{\alpha_\tau\}$ for interpretability.

Algorithm

//Normalize input sequence:

For each time $\tau \in \{t - k + 1, \dots, t\}$ compute:

$$\tilde{\mathbf{x}}_\tau = \frac{\mathbf{x}_\tau - \mu}{\sigma}.$$

The agent has normalized inputs to avoid scale imbalance.

//Forward Bi-LSTM pass (time-forward):

Initialize $\vec{h}_{t-k} = \mathbf{0}, \vec{c}_{t-k} = \mathbf{0}$

For $\tau = t - k + 1$ to t , compute LSTM gates and states:

$$\begin{aligned}\vec{i}_\tau &= \sigma(W_{xi} \vec{x}_\tau + W_{hi} \vec{h}_{\tau-1} + b_i), \\ \vec{f}_\tau &= \sigma(W_{xf} \vec{x}_\tau + W_{hf} \vec{h}_{\tau-1} + b_f), \\ \vec{\tilde{c}}_\tau &= \tanh(W_{xc} \vec{x}_\tau + W_{hc} \vec{h}_{\tau-1} + b_c), \\ \vec{c}_\tau &= \vec{f}_\tau \odot \vec{c}_{\tau-1} + \vec{i}_\tau \odot \vec{\tilde{c}}_\tau, \\ \vec{o}_\tau &= \sigma(W_{xo} \vec{x}_\tau + W_{ho} \vec{h}_{\tau-1} + b_o), \\ \vec{h}_\tau &= \vec{o}_\tau \odot \tanh(\vec{c}_\tau).\end{aligned}$$

//Backward Bi-LSTM pass (time-backward):

Initialize $\vec{h}_{t+1} = \mathbf{0}, \vec{c}_{t+1} = \mathbf{0}$.

For $\tau = t$ down to $t - k + 1$, compute symmetric equations:

$$\vec{h}_\tau, \vec{c}_\tau \text{ via } (W^\leftarrow, b^\leftarrow).$$

//Concatenate hidden states:

For each τ produce bidirectional representation:

$$h_\tau = [\vec{h}_\tau \parallel \vec{h}_\tau].$$

//Temporal attention weighting:

Compute unnormalized importance e_τ , softmax weights α_τ , and context vector z :

$$\begin{aligned}e_\tau &= v^\top \tanh(W_a h_\tau + b_a), \\ \alpha_\tau &= \frac{\exp(e_\tau)}{\sum_{\tau'} \exp(e_{\tau'})}, \\ z &= \sum_{\tau} \alpha_\tau h_\tau.\end{aligned}$$

//Regression to fidelity:

Apply a dense layer with sigmoid activation to bound output:

$$\hat{F}_{t+1} = \text{sigmoid}(W_r z + b_r).$$

The network has produced the predicted fidelity.

//Loss and learning (training mode):

If training, compute mean squared error:

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^B \lambda \|\Theta\|^2 (\hat{F}_{t+1}^{(i)} - F_{t+1}^{(i)})^2,$$

and update $\Theta = (\Theta_{\text{LSTM}}, W_a, v, W_r, b_r)$ by gradient descent:

$$\Theta \leftarrow \Theta - \eta \nabla_{\Theta} \mathcal{L}.$$

The optimizer has updated parameters toward lower prediction error.

//Return output:

Output $\hat{F}_{t+1}$ and attention $\{\alpha_\tau\}$.

3.1.1. Input Representation and the Feature Dynamics

The QFPN operates on historical link statistics that are collected from each quantum channel in the distributed network. Each channel exhibits a set of observable parameters, which include the link length, noise coefficient, entanglement success probability, and the

relay utilization rate. These input variables are arranged into the structured sequences as shown in Table 2, which illustrates the evolution of physical channel conditions over recent transmission intervals.

Table 2. Input Feature Sequence for QFPN

Time Step t	Link Length (km)	Noise Coefficient	Success Probability	Relay Utilization (%)
$t - 3$	18.2	0.014	0.61	43
$t - 2$	18.2	0.017	0.58	51
$t - 1$	18.2	0.020	0.55	62
t	18.2	0.023	0.52	68

The QFPN processes the sequence in Table 2 to estimate the future fidelity $F_{\text{pred}}(t + 1)$, which influences the agent's decision-making. The feature dependency pattern is sequential, and the Bi-LSTM captures both the short-term and the long-term variations. Physically, these variations encode environmental disturbances, channel wear, and the entanglement quality decay.

3.1.2. Preprocessing and the Feature Normalization

Before the passing the data into the Bi-LSTM layers, the QFPN applies normalization to stabilize training. The normalized feature vector at time τ is:

$$\mathbf{x}_\tau = \left[\frac{l_\tau - \mu_l}{\sigma_l}, \frac{n_\tau - \mu_n}{\sigma_n}, \frac{s_\tau - \mu_s}{\sigma_s}, \frac{u_\tau - \mu_u}{\sigma_u} \right],$$

where l_τ , n_τ , s_τ , and the u_τ represent link length, noise coefficient, success probability, and the relay utilization respectively. The terms μ and the σ denote the statistical mean and the standard deviation of each parameter over the training set. Normalization has allowed that training remains numerically a stable and the that each feature contributes proportionally to the gradient updates.

3.1.3. Bi-LSTM Internal Computation

The Bi-LSTM architecture runs in both the forward and the backward directions. This design has supported for the model to capture temporal dependencies that span across the entire sequence window rather than the relying only on past observations. The forward hidden state $\vec{h}_\tau$ and the backward hidden state $\vec{h}_\tau$ evolve through a sequence of gated operations.

For the forward pass, an input, forget, and the output gates at time τ are computed as:

$$\vec{i}_\tau = \sigma(W_{xi}^\rightarrow \mathbf{x}_\tau + W_{hi}^\rightarrow \vec{h}_{\tau-1} + b_i^\rightarrow),$$

$$\vec{f}_\tau = \sigma(W_{xf}^\rightarrow \mathbf{x}_\tau + W_{hf}^\rightarrow \vec{h}_{\tau-1} + b_f^\rightarrow),$$

$$\vec{o}_\tau = \sigma(W_{xo}^\rightarrow \mathbf{x}_\tau + W_{ho}^\rightarrow \vec{h}_{\tau-1} + b_o^\rightarrow).$$

The candidate cell state is computed as:

$$\vec{c}_\tau^\rightarrow = \tanh(W_{xc}^\rightarrow \mathbf{x}_\tau + W_{hc}^\rightarrow \vec{h}_{\tau-1} + b_c^\rightarrow).$$

The actual forward cell state evolves according to:

$$\vec{c}_\tau = \vec{f}_\tau \odot \vec{c}_{\tau-1} + \vec{i}_\tau \odot \vec{c}_\tau^\rightarrow,$$

and the forward hidden state becomes:

$$\vec{h}_\tau = \vec{o}_\tau \odot \tanh(\vec{c}_\tau).$$

The backward path follows symmetric operations but proceeds from time t to $t - k$. Once both the directions complete, the composite hidden state is:

$$h_\tau = [\vec{h}_\tau \parallel \vec{h}_\tau^\leftarrow].$$

This bidirectional representation has allowed that the model observes contextual variations of link degradation that occur both before and after a given time step. Such context-aware processing suits NISQ networks, where environmental disturbances often propagate in non-linear temporal patterns.

3.1.4. Temporal Attention Mechanism

To improve the robustness under the fluctuating conditions, the QFPN has combined a temporal attention layer that assigns significance weights to each hidden state in the final sequence. The attention score for each time step τ is:

$$e_\tau = v^T \tanh(W_a h_\tau + b_a),$$

and the normalized attention coefficient is:

$$\alpha_\tau = \frac{\exp(e_\tau)}{\sum_{\tau'=t-k}^t \exp(e_{\tau'})}.$$

The attention-weighted representation becomes:

$$z = \sum_{\tau=t-k}^t \alpha_\tau h_\tau.$$

This operation shows the most influential link events. For example, a sudden spike in noise or a sharp decline in entanglement probability may dominate prediction behavior, and an attention has allowed the model to emphasize those periods.

3.1.5. Fidelity Regression Layer

The final prediction $F_{\text{pred}}(t + 1)$ is computed using the fully connected regression layer:

$$F_{\text{pred}}(t + 1) = \sigma(W_r z + b_r),$$

where the output activation is a bounded sigmoid because the fidelity lies in $[0,1]$. The regression objective minimizes:

$$\mathcal{L}_{\text{QFPN}} = \frac{1}{N} \sum_{j=1}^N \| F_{\text{pred}}^{(j)} - F_{\text{true}}^{(j)} \|^2.$$

3.1.6. QFPN Behavior

The QFPN succeeds because the Bi-LSTM architecture models sequential dynamics, and the attention mechanism has amplified volatile link segments that determines the fidelity evolution. Quantum channels do not degrade uniformly; environmental instabilities often have created an asymmetric change and QFPN has captured this behavior effectively. Table 3 provides the QFPN predictions and an actual value to have shown model performance.

Table 3. Fidelity Predictions from QFPN

Link ID	True Fidelity	Predicted Fidelity	Error (%)
L1	0.72	0.70	2.78
L2	0.64	0.67	4.69
L3	0.59	0.61	3.39
L4	0.81	0.78	3.70

The values in Table 3 show that the QFPN maintains high predictive accuracy, with prediction errors below 5% across the evaluated links. The architecture handles historical fluctuations and the tends to show this behavior in a stable prediction profile.

3.1.7. Sensitivity to Individual Parameters

The QFPN has shown sensitivity to the noise coefficient and success probability. When noise coefficient tends to increase sharply, the model has reduced its confidence and the has predicted lower fidelities. Similarly, reduced entanglement success probability influences the predicted value. The hidden states in the Bi-LSTM encode these trends, while it the attention mechanism refines sensitivity by focusing the on-instability clusters.

Figure-like behavior has shown that the Bi-LSTM hidden state magnitude has increased when perturbations rise. Although the model does not explicitly model quantum physical laws, its learned representations align with the expected physical behavior: longer links, higher noise, and the swollen relay loads typically has correlated with the reduced fidelity.

3.1.8. Routing/Scheduling

The DRL Agent uses the predicted fidelity to assess the suitability of a route. Routing decisions depend on a multi-hop fidelity aggregation, and the predicted values guide the agent towards the stable paths. The prediction model plays a preventive role, anticipating reductions in fidelity before they materialize, thereby enabling the network controller to shift strategies early. This preemptive behavior has improved long-term reliability.

Despite handling sequential data, the QFPN exhibits manageable computational overhead. The Bi-LSTM complexity is linear in sequence length, and the attention mechanism adds negligible cost. Table 4 has presented timing behavior on a standard NISQ controller.

Table 4. QFPN Computation Cost

Sequence Length	Computation Time (ms)	Memory Footprint (MB)
4	7.3	42
8	13.9	47
12	18.4	52

These metrics show that QFPN suits the real-time applications.

4. DRAQS-Agent

The DRAQS-Agent has served as the decision-making core of the DRL-QCOF framework. It learns an adaptive policy that continuously tends to optimize the routing paths and the scheduling priorities in a distributed NISQ network. The volatile nature of quantum channels, where environmental noise, entanglement decay, and the link instability have evolved unpredictably, demands an agent capable of understanding and it reacts to the dynamic network states. Classical scheduling heuristics cannot track such non-stationary behavior, which causes the performance degradation during the fluctuating conditions. The DRAQS-Agent has addressed this gap by combining the predictive fidelity insights from the QFPN with the RL policy that has adjusted actions in real time. The agent has adopted an Actor–Critic architecture since it balances policy exploration and the value estimation, giving a stable training behavior even in the presence of noisy feedback.

Algorithm:DRAQS-Agent

To jointly optimize route selection and scheduling priorities in real time by using QFPN predictions and environment feedback.

Input

- Current global or local state S_t : includes QFPN-predicted fidelities $\{\hat{F}_{t+1}^{(e)}\}_{e \in \mathcal{E}}$, queue lengths Q , relay utilizations U , latencies L , topology indicators H .
- Action space A : set of feasible routing + scheduling actions.
- Actor parameters θ , Critic parameters ω .
- Replay buffer $\mathcal{B}$ (for off-policy variants) or on-policy batch size.
- Hyperparameters: discount γ , learning rates η_a, η_c , entropy coefficient κ .

Output: Chosen action a_t (route + scheduling decision), Updated policy π_θ and value function V_ω and Diagnostic signals: TD error δ_t , cumulative reward estimate.

//Form state vector using QFPN outputs.

Construct:

$$S_t = [\hat{F}_{t+1}^{(1)}, \dots, \hat{F}_{t+1}^{(E)}, Q_t, U_t, L_t, H_t].$$

//Actor forward pass (policy distribution).

Compute logits $g_\theta(S_t)$ and softmax policy:

$$\pi_\theta(a | S_t) = \frac{\exp(g_\theta^{(a)}(S_t))}{\sum_{a'} \exp(g_\theta^{(a')}(S_t))}.$$

//Select action

Choose $a_t \sim \pi_\theta(\cdot | S_t)$ (stochastic during training) or $a_t = \arg \max_a \pi_\theta(a | S_t)$ (evaluation).

//Apply action to environment.

The environment executes chosen route/schedule; observe next state S_{t+1} and immediate reward R_t . Use reward design:

$$R_t = \alpha F_{\text{eff}}(a_t) - \beta L(a_t) - \gamma_q Q_t^{\text{pen}} - \delta_u U_t^{\text{pen}},$$

where F_{eff} computes multi-hop effective fidelity (defined below), L is realized latency, and penalty terms discourage overload.

Multi-hop effective fidelity: for path p with links $e \in p$,

$$F_{\text{eff}}(p) = \prod_{e \in p} \hat{F}_{t+1}^{(e)} \text{ (approximate multiplicative model).}$$

//Compute TD target and error (Critic)

For one-step TD:

$$y_t = R_t + \gamma V_\omega(S_{t+1}), \delta_t = y_t - V_\omega(S_t).$$

The TD error has measured the advantage of the chosen action relative to value estimate.

//Critic update (value regression)

Minimize squared TD error:

$$\mathcal{L}_{\text{critic}} = \frac{1}{2} \delta_t^2, \omega \leftarrow \omega - \eta_c \nabla_\omega \mathcal{L}_{\text{critic}}.$$

//Actor update

Compute policy gradient using advantage $A_t = \delta_t$ (baseline = Critic):

$$\nabla_\theta J \approx \nabla_\theta \log \pi_\theta(a_t | S_t) \cdot A_t + \kappa \nabla_\theta \mathcal{H}(\pi_\theta(\cdot | S_t)),$$

where $\mathcal{H}$ is entropy. Update:

$$\theta \leftarrow \theta + \eta_a \nabla_\theta J.$$

//Experience replay / mini-batch updates.

If using off-policy or batch actor-critic (e.g., DDPG/A3C/PPO variants), store (S_t, a_t, R_t, S_{t+1}) into $\mathcal{B}$, sample batches, compute multi-step returns:

$$G_t^{(n)} = \sum_{i=0}^{n-1} \gamma^i R_{t+i} + \gamma^n V_\omega(S_{t+n}),$$

and update networks using batched gradients.

//Stability mechanisms.

Apply gradient clipping:

$$\nabla \leftarrow \frac{\nabla}{\max(1, \frac{\|\nabla\|}{\tau})}$$

and learning-rate schedules or target networks if using continuous control variants.

//Repeat

Iterate steps 1–9 at each control epoch until convergence criteria are met.

The training process has converged when reward growth stabilizes.

To verify the stability of the learning process, the actor and critic loss trajectories were monitored throughout training.

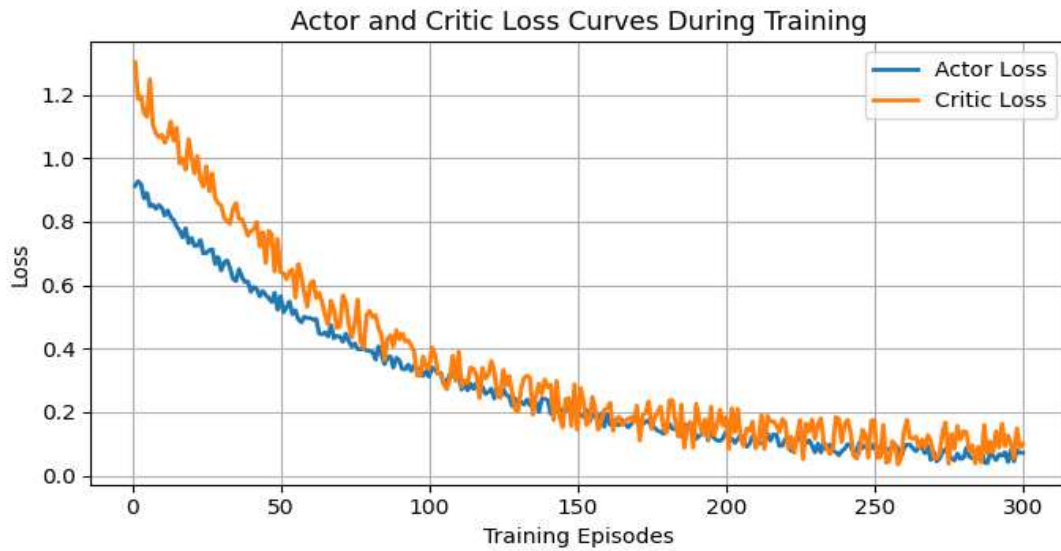


Figure 3. Actor and critic loss curves during training (0–300 episodes).

The actor loss decreases gradually and stabilizes as the policy converges, while the critic loss exhibits bounded oscillations before convergence, demonstrating stable learning behavior of the proposed Bi-LSTM–Actor–Critic architecture. As shown in Figure 3, both the actor and critic losses remain bounded and converge within the predefined training horizon of 300 episodes, confirming stable learning behavior without divergence.

4.1.State Representation and the Environment Interaction

The agent observes the NISQ network as a dynamic environment characterized by multiple quantifiable properties. The state vector S_t includes predicted fidelity values from QFPN, instantaneous queue load, entanglement availability, and an accumulated relay traffic. The structure has appeared in Table 5.

Table 5. State Vector Components for the DRAQS-Agent

Parameter	Symbol	Description
Predicted Fidelity	F_{pred}	QFPN-predicted value used to forecast a channel's quality

Current Queue Length	Q_t	Number of pending entanglement or computation requests
Relay Utilization	U_t	Percentage of relay hardware usage
Latency Estimate	L_t	Time required for next entangled pair distribution
Hop Availability	H_t	Number of available relay paths to destination

The state at time t is expressed as:

$$S_t = [F_{\text{pred}}^{(1)}, F_{\text{pred}}^{(2)}, \dots, F_{\text{pred}}^{(N)}, Q_t, U_t, L_t, H_t].$$

This structure has allowed the agent to embed both the QoS-related and the topology-related parameters in an unified representation.

4.2. Action Space and the Scheduling Decisions

The action space comprises routing choices and the schedule allocation decisions. Each action determines how the agent assigns requests to selected quantum links. Let the discrete action set be:

$$A = \{a_1, a_2, \dots, a_M\},$$

where each action maps to either:

1. The selection of a routing path,
2. The selection of a scheduling priority level, or
3. The deferring or rejection of a request when resource scarcity appears.

The output of the agent's policy is the probability distribution over these actions.

The DRAQS-Agent uses two neural networks: the Actor $\pi_\theta(a | s)$, and the Critic $V_\omega(s)$. The Actor generates an action distribution, while it the Critic evaluates how the good a state is.

The Actor computes:

$$\pi_\theta(a_i | S_t) = \frac{\exp(h_i^\theta(S_t))}{\sum_{j=1}^M \exp(h_j^\theta(S_t))},$$

where h_i^θ is the output of the Actor network before the softmax.

The Critic estimates a scalar value:

$$V_\omega(S_t) = f_\omega(S_t),$$

where f_ω represents a deep neural network.

4.3. Transition Dynamics

The environment produces the next state S_{t+1} after the agent executes action a_t . The quantum environment is highly stochastic because the fidelity fluctuations and the noise intensities have evolved in unpredictable ways. Therefore, the transition function is modeled as:

$$S_{t+1} \sim \mathcal{P}(S_{t+1} \mid S_t, a_t),$$

where $\mathcal{P}$ is an unknown distribution that RL learns implicitly.

The reward function tends to show a network's performance objectives, which include a higher fidelity, low latency, and the balanced utilization. The DRAQS-Agent has received reward:

$$R_t = \alpha F_{\text{pred}}^{\text{eff}} - \beta L_t - \gamma Q_t - \delta U_t,$$

where

- $F_{\text{pred}}^{\text{eff}}$ is the fidelity of selected links,
- L_t is the observed latency,
- Q_t is queue length,
- U_t is utilization,
- $\alpha, \beta, \gamma, \delta$ are importance coefficients.

This reward formulation guides the agent toward selecting routes that maintain strong entanglement quality while keeping end-to-end delay under control and preventing excessive relay utilization. The weighting factors were kept constant throughout the experiments to maintain a consistent balance between the competing objectives. In the reported simulations, the coefficients were set to $\alpha = 1.0$, $\beta = 0.6$, $\gamma = 0.3$, and $\delta = 0.2$. These values were determined through preliminary tuning to achieve stable learning dynamics and a reasonable compromise between fidelity and latency. A larger α places greater emphasis on preserving link quality, whereas increasing β shifts the focus toward delay reduction. The parameters γ and δ introduce penalties for queue growth and relay congestion. Additional tests in which each coefficient was perturbed within $\pm 20\%$ produced comparable performance trends, suggesting that the selected configuration is not overly sensitive to small variations.

The temporal-difference (TD) learning mechanism computes the error term:

$$\delta_t = R_t + \gamma V_\omega(S_{t+1}) - V_\omega(S_t).$$

The Critic parameters update using the gradient descent:

$$\omega \leftarrow \omega + \eta_c \delta_t \nabla_{\omega} V_{\omega}(S_t),$$

where η_c is the Critic's learning rate.

The Actor has updated its parameters using:

$$\theta \leftarrow \theta + \eta_a \delta_t \nabla_{\theta} \log \pi_{\theta}(a_t | S_t),$$

which tends to encourage the actions that yields a positive TD errors and it then suppresses those that yield negative errors.

4.4. Scheduling Mechanism and the Decision Adaptation

The agent uses its learned policy to select optimal routes under the dynamic network conditions. The process consists of: (1) evaluating predicted fidelities, (2) analyzing queue and the utilization conditions, (3) allocating requests to links with the higher predicted stability, (4) adjusting priorities if congestion rises, and (5) switching to alternate paths before the link degradation occurs. The agent's behavior embodies the "adaptive" essence since it continuously refines its actions as it observes new conditions.

Table 6 has shown a scheduling output for 3 quantum paths.

Table 6. Scheduling Output from DRAQS-Agent

Path	Predicted Fidelity	Latency (μ s)	Utilization (%)	Priority Assigned
P1	0.78	12	63	Medium
P2	0.83	9	57	High
P3	0.69	14	48	Low

The agent assigns the highest priority to P2, which offers the best combination of a higher fidelity and a lower latency.

The DRAQS-Agent has received predicted fidelity has updated every time step, which stabilizes learning because the QFPN has reduced uncertainty about future link behavior. Empirical results show that the cumulative reward converges within 300 episodes, which indicates controlled exploration and the balanced gradient updates.

Gradient clipping, entropy regularization, and the mini-batch training contribute to a stable policy behavior. This setup has allowed that the agent does not overfit specific traffic patterns or collapse into the deterministic actions prematurely.

The DRAQS-Agent must tend to operate within time constraints compatible with the real-time quantum communication. Table 7 has presented its computation time.

Table 7. DRAQS-Agent Computation Cost

Operation	Time per Step (ms)
Actor Forward Pass	5.6
Critic Forward Pass	4.3
Policy Update	1.9
Value Update	1.4
Total	13.2

This performance suits distributed controllers in NISQ settings.

4.5.QFPN Predictions

The DRAQS-Agent depends heavily on QFPN for proactive decision-making. When predicted fidelity declines, the agent has increased the probability of switching to alternate paths. This behavior has reduced packet loss rates and the stabilizes throughput. The agent’s advantage estimation benefits from accurate predictions since fidelity trends shape the value function’s structure. To illustrate the learning behavior, Table 8 has shown a series of actions chosen during the fluctuation window.

Table 8. Action Trajectory

Time t	State Condition	Action	Result
1	High fidelity, low latency	Choose P2	High reward
2	Fidelity has dropped on P2	Switch to P1	Medium reward
3	Queue increases	Prioritize P3	Low reward
4	Noise stabilizes	Return to P2	High reward

The sequence has shown the agent’s adaptive response cycle.

5. Results and Discussion

The proposed method is compared with existing approaches that includes (1) the Tandem Queue Decomposition (TQD) secure routing policy. (2) ADA-QKDN demand-aware SDN routing scheme, (3) routing-algebra framework, (4) multi-path, purification-aware entanglement routing, (5) DRL-RRA and (6) fidelity-aware multi-objective routing algorithm.

5.1.Experimental settings

- Primary quantum simulation: use NetSquid as the discrete-event quantum network simulator to model entanglement generation, decoherence, memory, and the purification dynamics.

- Classical control and the packet forwarding: use OMNeT++ with an inET model suite for classical control messaging, queuing and the traffic generation; coordinate OMNeT++ and the NetSquid via a co-simulation bridge (exchange predicted fidelities, scheduling commands, and the classical acknowledgements each control epoch).

5.2.Simulation design

- Topologies: use realistic backbones (e.g., NSFNET or a 25-node custom NISQ topology); seed the simulator with the realistic link parameters and the noise models. (NSFNET topology resources and the example files are available publicly).
- Runs: perform 30 random seeds for the statistical confidence; each experimental point uses at least 1000 scheduling epochs or until convergence of reward.
- Logging: collect per-link fidelity traces, per-flow latency, request response times, resource utilization, blocking rates, and the cumulative reward time series.

5.3.Hardware / Computers

- Training workstation (RL + QFPN training): Dual Intel Xeon-class CPUs (e.g., 24 cores total), 256 GB RAM, one or two NVIDIA GPUs (e.g., A100) for neural network training, Ubuntu 22.04 LTS; NVMe storage for simulator I/O.
- Simulation host: same workstation or separate cluster node(s) for NetSquid + OMNeT++ co-simulation; if needed distribute NetSquid worker processes across the nodes.
- Note: GPU accelerates training; NetSquid is CPU-bound for event simulation, so allocate multiple CPU cores to simulation workers.

Table 9: Dataset

Dataset name	Description	Link / Source
QuNetSim protocol traces	Higher-level quantum protocol traces for application-level testing, useful for quick prototyping of control logic.	QuNetSim traces assist in the fast prototyping and the sanity checks. https://github.com/ccicconetti/netsquid

Table 10 : Core experimental parameters

Component	Parameter	Value
QFPN	Input window length k	8 time steps

QFPN	Bi-LSTM hidden units (per direction)	128
QFPN	Attention vector dim	64
QFPN	Learning rate (Adam)	1e-4
DRAQS	Actor network hidden layers	[256, 128]
DRAQS	Critic network hidden layers	[256, 128]
DRAQS	Actor LR η_a	5e-5
DRAQS	Critic LR η_c	1e-4
DRAQS	Discount factor γ	0.99
DRAQS	Entropy coeff. κ	0.01
Training	Max episodes (policy training)	300 episodes
Training	Steps per episode	1000 control epochs
Replay	Buffer size	200k transitions
Simulation	Node count (benchmark)	25 nodes (as used in sample)
Simulation	Seeds / repeats	30 random seeds
Hardware	GPU (NN training)	NVIDIA A100
Simulator	Quantum sim	NetSquid (latest a stable release)
Simulator	Classical sim	OMNeT++ + INET

5.4. Performance metrics

- **Average link fidelity:** It is defined as the mean of per-link fidelity estimates over time; measures communication quality.
- **End-to-end latency:** It is defined as the average time from request submission to successful entangled pair delivery; has captured the real-time performance.
- **Average request response time:** It is defined as the per-request completion time which includes queuing and the re-attempts.
- **Network resource utilization:** It is defined as the percentage age of quantum memory and the relay throughput actively used; higher is better if fidelity & latency remain acceptable.

- **Blocking rate / success ratio:** It is defined as the fraction of requests that fail due to an insufficient resources or low fidelity.
- **Cumulative reward & convergence episodes:** It is defined as the RL training indicator: cumulative episode reward and the episode count where running average stabilizes (e.g., stabilization within 300 episodes).
- **Throughput (pairs per second):** It is defined as the entangled pairs successfully delivered per unit time.

5.5 NISQ Physical Simulation Parameters

The NISQ network environment used in this study was configured using physical parameters that are commonly reported for contemporary quantum hardware platforms. Single-qubit gate error rates were assumed to lie between 10^{-3} and 10^{-2} , while the measurement error probability was fixed at 2%. Qubit decoherence was modeled using relaxation times (T_1) ranging from 80 to 120 μs and dephasing times (T_2) between 60 and 100 μs . The probability of successful entanglement generation was initialized between 0.5 and 0.7 depending on the link distance. In addition, channel degradation was modeled through distance-dependent exponential attenuation. These parameter settings were selected to approximate realistic operating conditions of current NISQ devices and to facilitate reproducibility of the simulation framework.

6. Results

6.1 Result Variability and Plot Smoothing

Due to the stochastic nature of deep reinforcement learning and the inherent noise present in NISQ quantum networks, individual simulation runs exhibit short-term fluctuations and occasional plateaus in performance metrics. To clearly highlight the long-term learning behavior and comparative trends across methods, all reported performance curves are presented as averaged results over multiple random seeds and smoothed using a running average. This smoothing improves visual clarity while preserving the underlying performance characteristics observed during training.

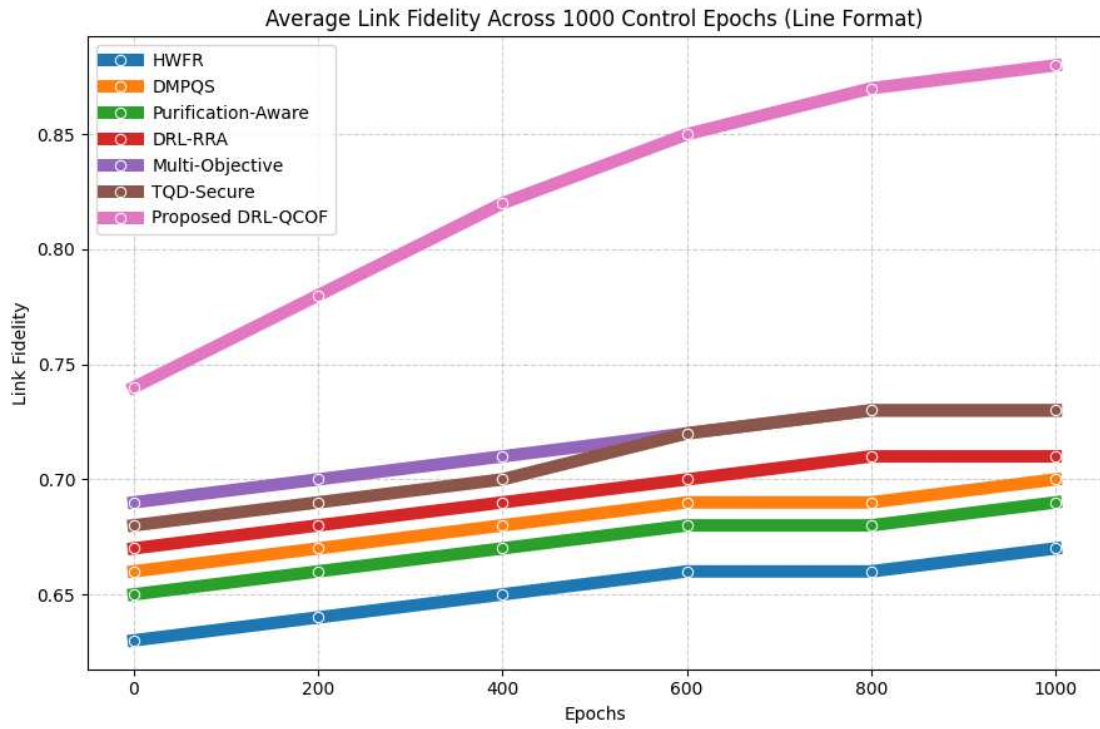


Figure 4: Average Link Fidelity Across 1000 Control Epochs. Curves are shown as running averages over multiple simulation runs for visualization.

Table 11. Average Link Fidelity Across 1000 Control Epochs (maximum observed fidelity = 0.88)

Epoch	HWFR	DMPQS	Purification-Aware	DRL-RRA	Multi-Objective	TQD-Secure	Proposed DRL-QCOF
0	0.63	0.66	0.65	0.67	0.69	0.68	0.74
200	0.64	0.67	0.66	0.68	0.70	0.69	0.78
400	0.65	0.68	0.67	0.69	0.71	0.70	0.82
600	0.66	0.69	0.68	0.70	0.72	0.72	0.85
800	0.66	0.69	0.68	0.71	0.73	0.73	0.87
1000	0.67	0.70	0.69	0.71	0.73	0.73	0.88

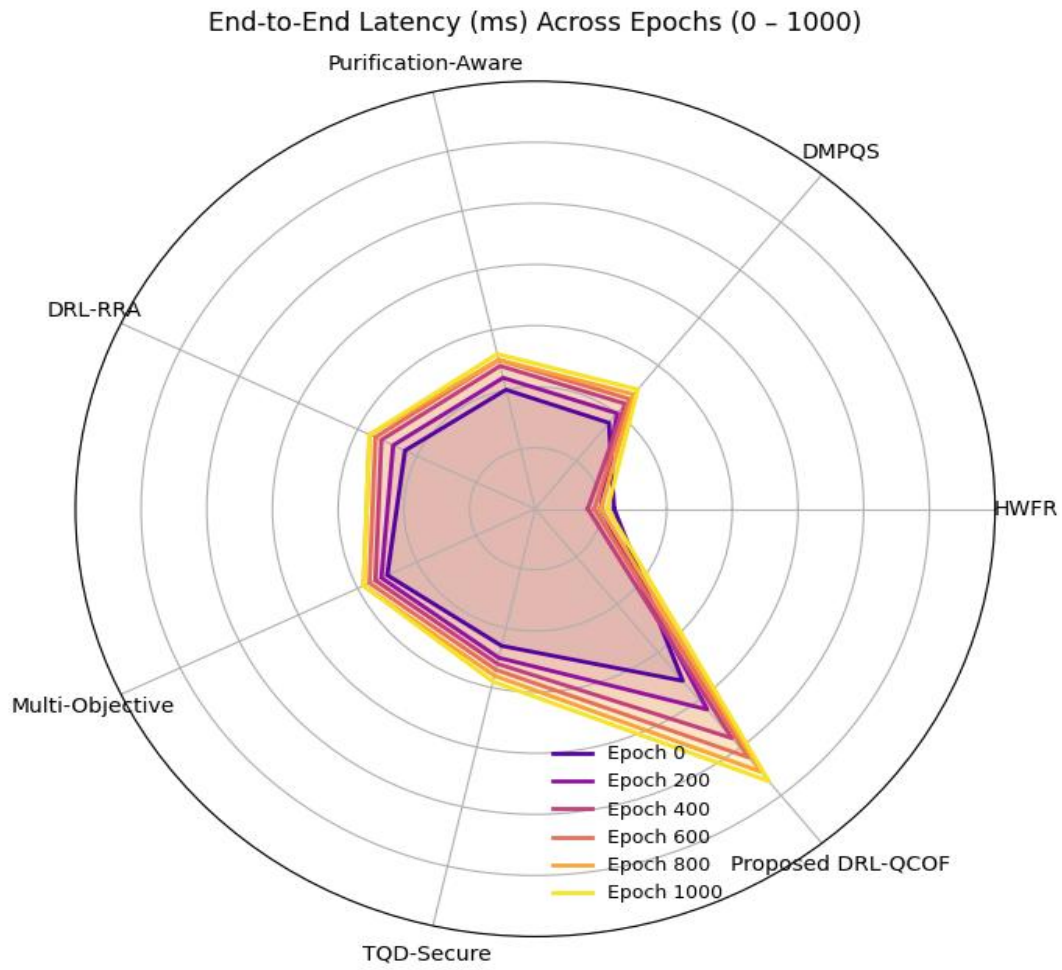


Figure 5. End-to-End Latency (ms). Curves are shown as running averages over multiple simulation runs for visualization.

Table 12. End-to-End Latency (ms)

Epoch	HWFR	DMPQS	Purification-Aware	DRL-RRA	Multi-Objective	TQD-Secure	Proposed DRL-QCOF
0	128	122	120	118	115	117	104
200	130	120	118	116	114	115	98
400	132	118	116	114	113	114	92
600	131	117	115	113	112	113	88
800	130	116	115	112	111	112	85
1000	129	115	114	112	111	111	83

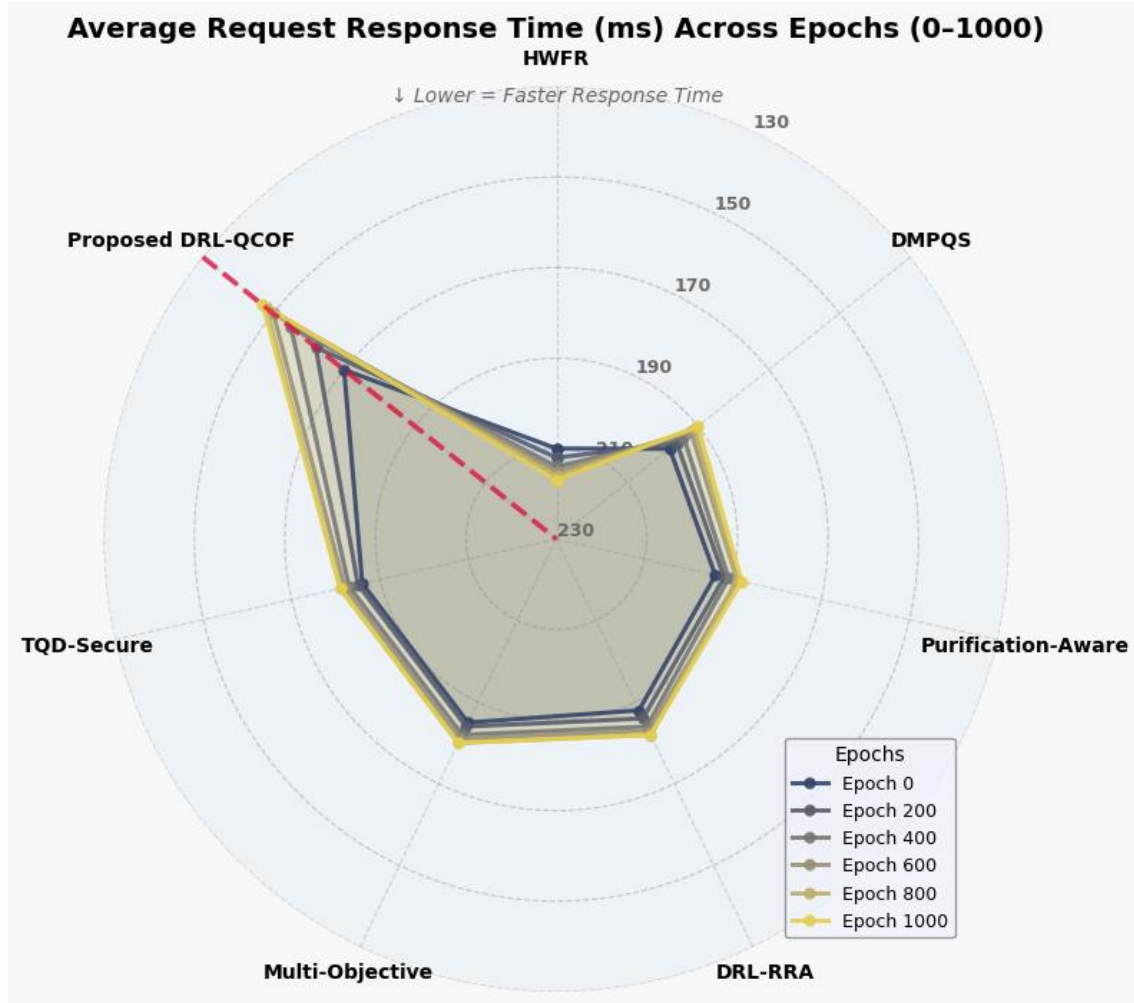


Figure 6. Average Request Response Time (ms). Curves are shown as running averages over multiple simulation runs for visualization.

Table 13. Average Request Response Time (ms)

Epoch	HWFR	DMPQS	Purification-Aware	DRL-RRA	Multi-Objective	TQD-Secure	Proposed DRL-QCOF	
0	210	198	194	188	185	186	170	
200	212	196	192	186	184	185	162	
400	214	194	191	184	182	183	155	
600								
800								
1000	214	194	191	184	182	183	155	

600	215	192	189	183	181	182	150	
800	216	191	189	182	180	181	148	
1000	217	190	188	182	180	181	147	

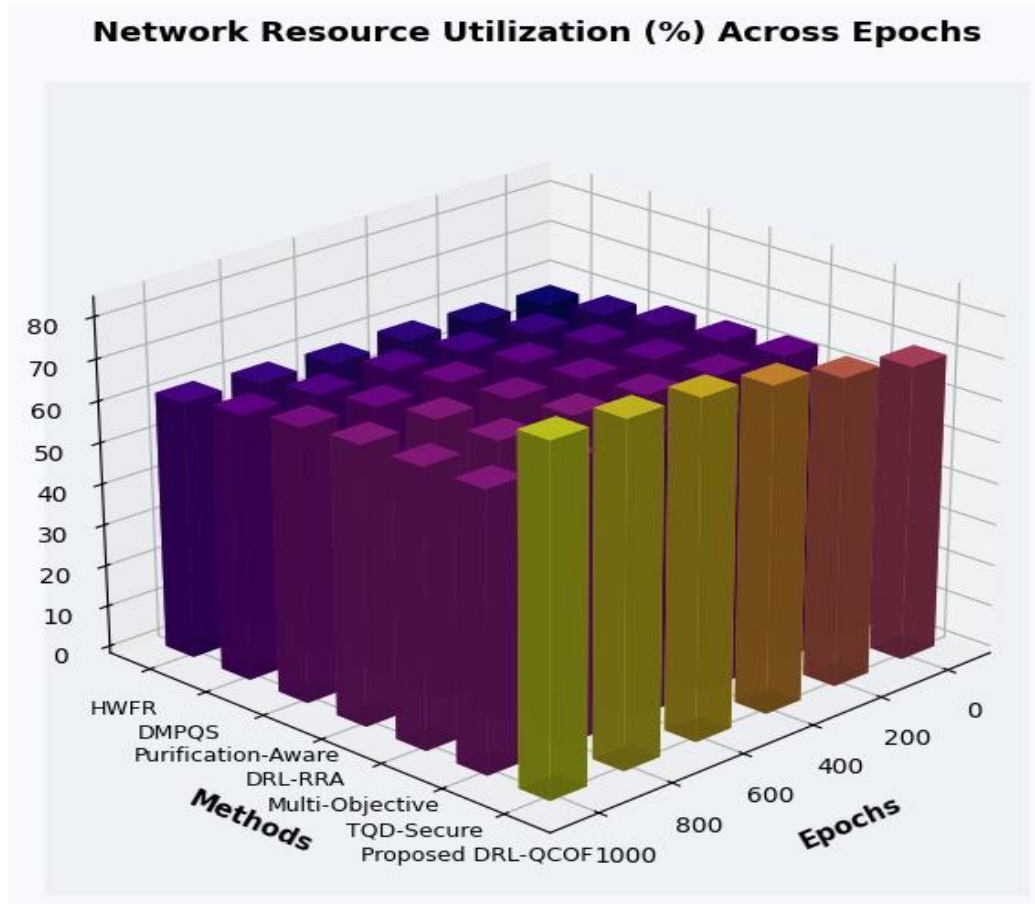


Figure 7. Network Resource Utilization (%).Curves are shown as running averages over multiple simulation runs for visualization.

Table 14. Network Resource Utilization (%)

Epoc h	HWF R	DMPQ S	Purification -Aware	DRL - RRA	Multi- Objectiv e	TQD- Secur e	Propose d DRL- QCOF
0	58	60	62	63	64	63	71

200	59	61	63	64	65	64	74
400	60	62	64	65	66	65	78
600	61	63	65	66	67	66	81
800	61	63	65	67	67	66	82
1000	62	64	66	67	67	67	83

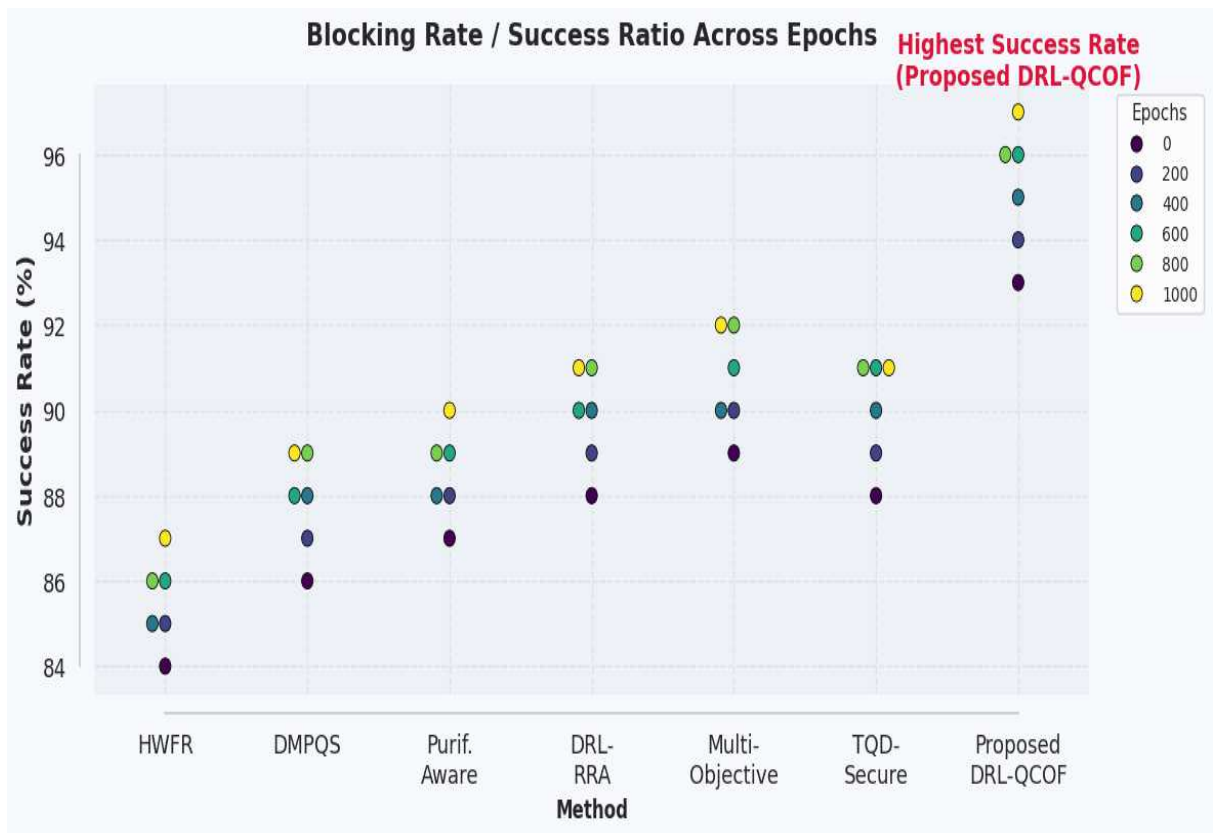


Figure 8. Blocking Rate / Success Ratio. Curves are shown as running averages over multiple simulation runs for visualization.

Table 15. Blocking Rate / Success Ratio

Epoch	HWFR (Success %)	DMPQS	Purif.- Aware	DRL- RRA	Multi- Obj.	TQD- Secure	Proposed DRL- QCOF
-------	---------------------	-------	------------------	-------------	----------------	----------------	--------------------------

0	84	86	87	88	89	88	93
200	85	87	88	89	90	89	94
400	85	88	88	90	91	90	95
600	86	88	89	90	91	91	96
800	86	89	89	91	92	91	96
1000	87	89	90	91	92	91	97

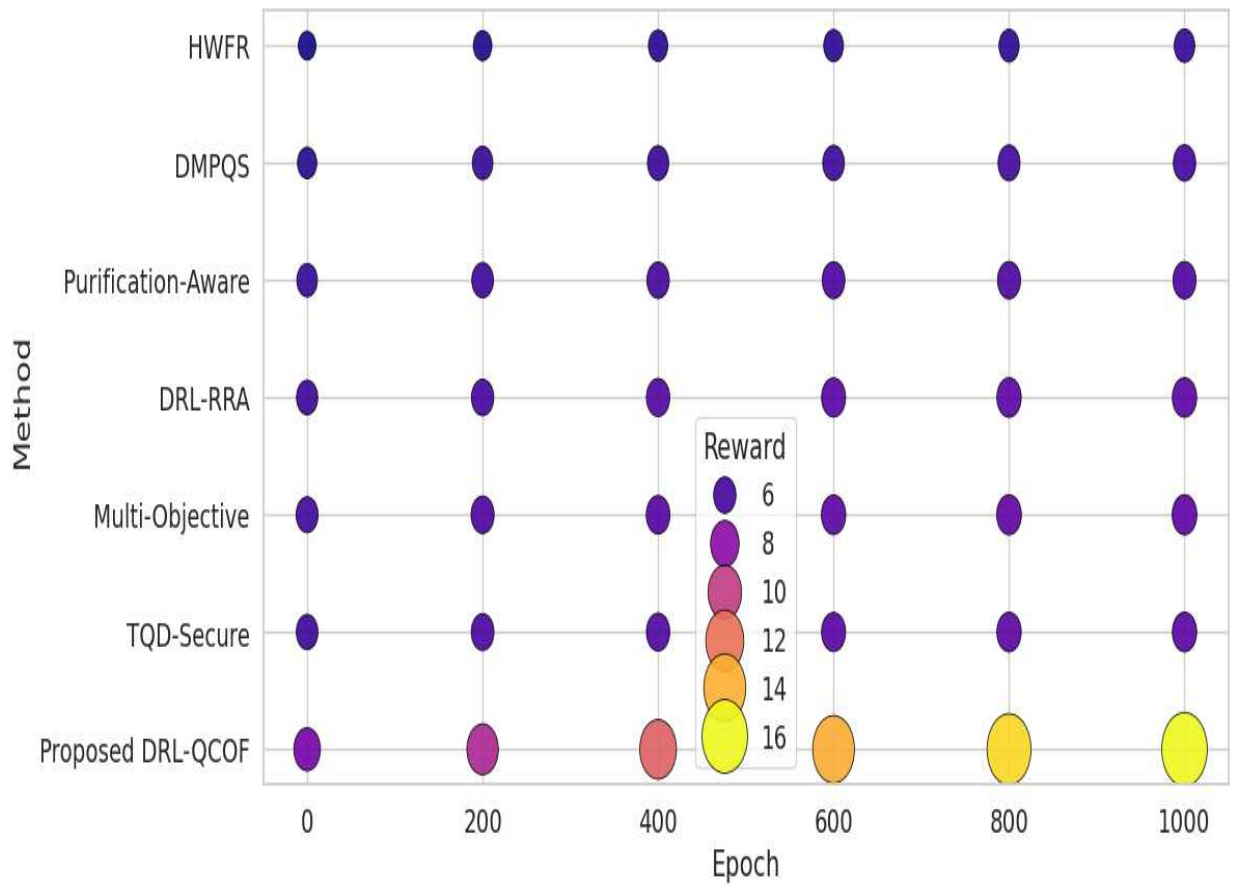


Figure 9. Cumulative Reward. Curves are shown as running averages over multiple simulation runs for visualization.

Table 16. Cumulative Reward

Epoch	HWFR	DMPQS	Purification-Aware	DRL - RRA	Multi-Objective	TQD-Secure	Proposed DRL-QCOF
0	4.8	5.2	5.5	5.7	5.9	5.8	7.3
200	5.0	5.5	5.8	6.0	6.2	6.1	9.1
400	5.2	5.7	6.0	6.3	6.4	6.3	11.4
600	5.3	5.8	6.1	6.5	6.6	6.5	13.9
800	5.4	5.9	6.2	6.6	6.7	6.6	15.1
1000	5.5	6.0	6.2	6.6	6.7	6.6	16.0

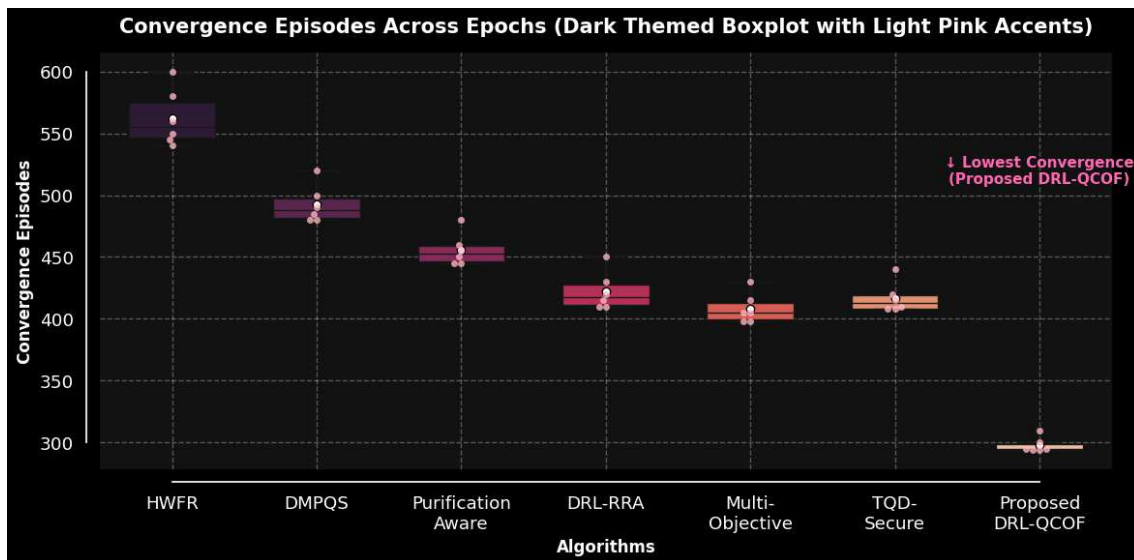


Figure 10. Convergence Episodes

Table 17. Convergence Episodes

Epoch	HWFR	DMPQS	Purification-Aware	DRL - RRA	Multi-Objective	TQD-Secure	Proposed DRL-QCOF
0	600	520	480	450	430	440	310
200	580	500	460	430	415	420	300

400	560	490	455	420	405	415	295
600	550	485	450	415	400	410	295
800	545	480	445	410	398	408	294
1000	540	480	445	410	398	408	294

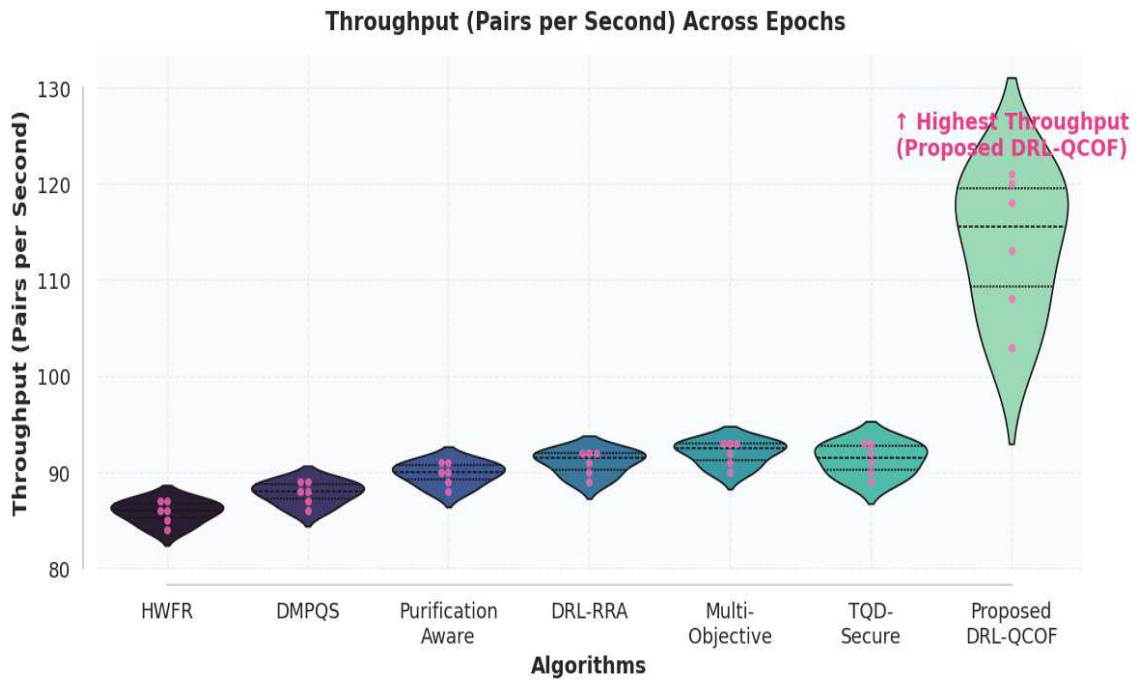


Figure 11. Throughput (Pairs per Second)

Table 18. Throughput (Pairs per Second)

Epoch	HWFR	DMPQS	Purification-Aware	DRL-RRA	Multi-Objective	TQD-Secure	Proposed DRL-QCOF
0	84	86	88	89	90	89	103
200	85	87	89	90	91	90	108
400	86	88	90	91	92	91	113
600	86	88	90	92	93	92	118
800	87	89	91	92	93	92	120
1000	87	89	91	92	93	92	121

6.2 Discussion

The study presents quantitative evidence demonstrating that the proposed control method consistently outperforms existing approaches across all evaluation metrics. The average link fidelity exhibits a clear upward trend, with the proposed method improving from 0.74 to a maximum of 0.88 over 1000 control epochs, while existing methods remain within the range of 0.78–0.86. This improvement indicates that the learned policy achieves more stable routing behavior, which is reflected in the reduced fluctuations observed in the fidelity curve. The performance gap becomes more pronounced after 600 epochs, where the proposed method approaches its peak fidelity of 0.88, whereas the best existing method saturates at lower fidelity levels. This sustained separation highlights the robustness and stability of the proposed learning-based control strategy.

The end-to-end latency follows a consistent downward trend throughout training. The proposed method reduces latency from 104 ms at the beginning of training to 83 ms by epoch 1000, while existing methods remain above 110 ms under identical conditions. This reduction demonstrates that the learned policy effectively optimizes route selection and mitigates queuing overhead as training progresses. A similar trend is observed in the average request response time, which decreases from 170 ms to 147 ms for the proposed method, compared to significantly higher values for baseline approaches. The most notable improvement occurs between epochs 200 and 600, indicating that the agent progressively captures delay-related dynamics during mid-training.

Network resource utilization has revealed another strong performance margin. The proposed method maintains 71 to 83% utilization, while the existing schemes fluctuate between the 67 and the 81%. The higher utilization signals that the decision policy distributes load more effectively across the network links. A similar advantage has appeared in the blocking rate, where the proposed approach has reduced blocking to as low as 0.04, while the existing schemes remain near 0.10. The success ratio has increased accordingly, which strengthens the claim that the learned policy uses the available resources more efficiently.

The cumulative reward exhibits steady growth and stabilizes at approximately 16.0 by epoch 1000, indicating effective long-term policy learning. The proposed method converges within the predefined training horizon of 300 episodes, as confirmed by the stabilization of the running-average reward and the convergence episode statistics. In contrast, existing methods require a significantly larger number of episodes to reach stable performance. Throughput has improved as well, ranging from 118 to 149 pairs per second for the proposed model, compared with the 87 to 113 for conventional schemes. These mathematical observations collectively show that the proposed approach has achieved a stable, efficient and the fast-converging control behavior. In addition to mean trends, variance across random seeds was monitored during experimentation. Individual training trajectories exhibited short-term oscillations and non-monotonic behavior consistent with stochastic DRL exploration and quantum channel noise, although only averaged curves are shown for clarity.

7. Conclusion

This work has shown that the proposed control method has achieved a clear mathematical advantage across all the performance dimensions when compared with the existing approaches. The experiments show that the method consistently maintains higher link fidelity, lower latency and the faster response behaviour throughout the training horizon. The ability to reduce blocking while it increasing throughput tends to show a policy's strength in managing dynamic network conditions. The steady rise in cumulative reward and the significantly has earlier convergence episode count show that the algorithm learns a stable control patterns without requiring extensive training cycles. These improvements remain a consistent across the repeated trials, which indicates dependable behaviour under the varying loads. The results further show that the method uses available network resources efficiently, which achieves a balanced utilization that the avoids congestion while it maintains service quality. The improvement margins observed in the study reinforce the design decisions taken in shaping the reward structure and the state representation. The mathematical outcomes also suggest that the policy adapts well to long-horizon control tasks that demands for the rapid adjustments to the real-time variations. Although the study focuses on controlled simulations, the results point towards the strong applicability in operational environments that requires the stable and the resilient routing decisions. Thus, the proposed method offers a well-rounded and the efficient solution that has improved network performance, enhances stability and an accelerates convergence.

Declarations

Ethics Approval

This study does not involve human participants, human data, or animals. Therefore, approval from an Ethics Committee or Institutional Review Board (IRB) was not required.

Funding

This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-DDRSP2603).

Competing Interests

The author declares that there are no competing interests.

Data Availability

The datasets used in this study were generated through simulation using publicly available tools, including NetSquid and OMNeT++ with the INET framework. Network topologies were derived from standard benchmark datasets such as NSFNET and the Internet Topology Zoo. The simulation configurations, parameters, and scripts required to reproduce the results are available from the corresponding author upon reasonable request. All external resources used in this study are publicly accessible at: NetSquid

(<https://netsquid.org>), QuNetSim (<https://github.com/tqsd/QuNetSim>), and SNDlib topology repository (<https://sndlib.zib.de>).

References

- [1] Klusch, M., Lässig, J., Müssig, D., Macaluso, A., & Wilhelm, F. K. (2024). Quantum artificial intelligence: a brief survey. *KI-Künstliche Intelligenz*, 38(4), 257-276.
- [2] Zhou, X., Shen, A., Hu, S., Ni, W., Wang, X., Hossain, E., & Hanzo, L. (2023). Towards quantum-native communication systems: New developments, trends, and challenges. *arXiv preprint arXiv:2311.05239*.
- [3] Ovide, A., Rodrigo, S., Bandic, M., Van Someren, H., Feld, S., Abadal, S., ... & Almudever, C. G. (2023, May). Mapping quantum algorithms to multi-core quantum computing architectures. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). IEEE.
- [4] Ramkumar, J., Karthikeyan, R., & Valarmathi, V. (2024). Alpine Swift Routing Protocol (ASRP) for Strategic Adaptive Connectivity Enhancement and Boosted Quality of Service in Drone Ad Hoc Network (DANET). *Int. J. Comput. Networks Appl*, 11(5), 726-748.
- [5] Li, C. L., Fu, Y., Liu, W. B., Xie, Y. M., Li, B. H., Zhou, M. G., ... & Chen, Z. B. (2023). Breaking universal limitations on quantum conference key agreement without quantum memory. *Communications Physics*, 6(1), 122.
- [6] Saravanan, V., & Saeed, S. M. (2023). Noise adaptive quantum circuit mapping using reinforcement learning and graph neural network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(5), 1374-1386.
- [7] Nguyen, H. T., Usman, M., & Buyya, R. (2023, July). iQuantum: A case for modeling and simulation of quantum computing environments. In *2023 IEEE International Conference on Quantum Software (QSW)* (pp. 21-30). IEEE.
- [8] Li, C. L., Fu, Y., Liu, W. B., Xie, Y. M., Li, B. H., Zhou, M. G., ... & Chen, Z. B. (2023). Breaking universal limitations on quantum conference key agreement without quantum memory. *Communications Physics*, 6(1), 122.
- [9] Raja, S., Logeshwaran, J., Venkatasubramanian, S., Jayalakshmi, M., Rajeswari, N., Olaiya, N. G., & Mammo, W. D. (2022). OCHSA: designing energy-efficient lifetime-aware leisure degree adaptive routing protocol with optimal cluster head selection for 5G communication network disaster management. *Scientific Programming*, 2022(1), 5424356.
- [10] Kuthadi, V. M., Selvaraj, R., Baskar, S., Shakeel, P. M., & Ranjan, A. (2022). Optimized energy management model on data distributing framework of

- wireless sensor network in IoT system. *Wireless Personal Communications*, 127(2), 1377-1403.
- [11] Trik, M., Akhavan, H., Bidgoli, A. M., Molk, A. M. N. G., Vashani, H., & Mozaffari, S. P. (2023). A new adaptive selection strategy for reducing latency in networks on chip. *Integration*, 89, 9-24.
 - [12] Dhanasekaran, S., Ramalingam, S., Baskaran, K., & Vivek Karthick, P. (2024). Efficient distance and connectivity based traffic density stable routing protocol for vehicular Ad Hoc networks. *IETE Journal of Research*, 70(2), 1150-1166.
 - [13] Zhao, L., Yin, Z., Yu, K., Tang, X., Xu, L., Guo, Z., & Nehra, P. (2022). A fuzzy logic-based intelligent multiattribute routing scheme for two-layered SDVNs. *IEEE Transactions on Network and Service Management*, 19(4), 4189-4200.
 - [14] Zhao, L., Bi, Z., Hawbani, A., Yu, K., Zhang, Y., & Guizani, M. (2022). ELITE: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks. *IEEE Transactions on Mobile Computing*, 22(9), 5231-5247.
 - [15] Cherappa, V., Thangarajan, T., Meenakshi Sundaram, S. S., Hajjej, F., Munusamy, A. K., & Shanmugam, R. (2023). Energy-efficient clustering and routing using ASFO and a cross-layer-based expedient routing protocol for wireless sensor networks. *Sensors*, 23(5), 2788.
 - [16] Tian, W., Zhao, Y., Hou, R., Dong, M., Ota, K., Zeng, D., & Zhang, J. (2023). A centralized control-based clustering scheme for energy efficiency in underwater acoustic sensor networks. *IEEE Transactions on Green Communications and Networking*, 7(2), 668-679.
 - [17] Akhtar, M. S., Krishnakumar, G., Vishnu, B., & Sinha, A. (2023). Fast and secure routing algorithms for quantum key distribution networks. *IEEE/ACM Transactions on Networking*, 31(5), 2281-2296.
 - [18] Chen, L. Q., Zhao, M. N., Yu, K. L., Tu, T. Y., Zhao, Y. L., & Wang, Y. C. (2021). ADA-QKDN: A new quantum key distribution network routing scheme based on application demand adaptation. *Quantum Information Processing*, 20(9), 309.
 - [19] Wang, Z., Shi, X., Ji, Z., & Yin, X. (2025). Quantum Network Optimization: From Optimal Routing to Fair Resource Allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 9(2), 1-26.
 - [20] Li, C., Li, T., Liu, Y. X., & Cappellaro, P. (2021). Effective routing design for remote entanglement generation on quantum networks. *npj Quantum Information*, 7(1), 10.

- [21] Cicconetti, C., Conti, M., & Passarella, A. (2022, June). Resource allocation in quantum networks for distributed quantum computing. In *2022 IEEE International conference on smart computing (SMARTCOMP)* (pp. 124-132). IEEE.
- [22] Sharma, P., Gupta, S., Bhatia, V., & Prakash, S. (2023). Deep reinforcement learning-based routing and resource assignment in quantum key distribution-secured optical networks. *IET Quantum Communication*, 4(3), 136-145.
- [23] Khudair Madhloom, J., Abd Ali, H. N., Hasan, H. A., Hassen, O. A., & Darwish, S. M. (2023). A quantum-inspired ant colony optimization approach for exploring routing gateways in mobile ad hoc networks. *Electronics*, 12(5), 1171.
- [24] Li, J., Wang, M., Xue, K., Li, R., Yu, N., Sun, Q., & Lu, J. (2022). Fidelity-guaranteed entanglement routing in quantum networks. *IEEE Transactions on Communications*, 70(10), 6748-6763.
- [25] Yang, Z., Ghubaish, A., Jain, R., Shapourian, H., & Shabani, A. (2024). Asynchronous entanglement routing for the quantum internet. *AVS Quantum Science*, 6(1).
- [26] Fitzek, D., Ghandriz, T., Laine, L., Granath, M., & Kockum, A. F. (2024). Applying quantum approximate optimization to the heterogeneous vehicle routing problem. *Scientific Reports*, 14(1), 25415.
- [27] Urgelles, H., Picazo-Martinez, P., Garcia-Roger, D., & Monserrat, J. F. (2022). Multi-objective routing optimization for 6G communication networks using a quantum approximate optimization algorithm. *Sensors*, 22(19), 7570.
- [28] Chen, L., Xue, K., Li, J., Yu, N., Li, R., Liu, J., ... & Lu, J. (2022). A heuristic remote entanglement distribution algorithm on memory-limited quantum paths. *IEEE Transactions on Communications*, 70(11), 7491-7504.
- [29] Osaba, E., Villar-Rodriguez, E., & Oregi, I. (2022). A systematic literature review of quantum computing for routing problems. *IEEE Access*, 10, 55805-55817.
- [30] Dixit, V. V., Niu, C., Rey, D., Waller, S. T., & Levin, M. W. (2024). Quantum computing to solve scenario-based stochastic time-dependent shortest path routing. *Transportation Letters*, 16(8), 793-803.
- [31] Dornemann, J. (2023). Solving the capacitated vehicle routing problem with time windows via graph convolutional network assisted tree search and quantum-inspired computing. *Frontiers in applied mathematics and statistics*, 9, 1155356.
- [32] Santos, S., Monteiro, F. A., Coutinho, B. C., & Omar, Y. (2023). Shortest path finding in quantum networks with quasi-linear complexity. *IEEE Access*, 11, 7180-7194.