A Digital Twin-Based Deep Reinforcement Learning Framework for Adaptive Scheduling in 5G/6G Networks

Naima Mchiri
LIMTIC Laboratory
University of Tunis El Manar
Ariana, Tunisia
naima.mchiri@fst.utm.tn

Rachid Zagrouba
LIMTIC Laboratory
University of Tunis El Manar
Ariana, Tunisia
rachid.zagrouba@isi.utm.tn

Ezzeddine Zagrouba
LIMTIC Laboratory
University of Tunis El Manar
Ariana, Tunisia
ezzedine.zagrouba@isi.utm.tn

***Abstract*— The transition toward AI-native 6G networks requires intelligent, adaptive, and reliable control mechanisms capable of handling highly dynamic and heterogeneous environments. In this context, reinforcement learning has emerged as a promising approach for optimizing network performance. However, existing works often focus on algorithmic design while overlooking critical aspects such as experimental rigor, reward formulation, and reproducibility, which can significantly impact the validity of the results. This paper proposes a Digital Twin-based Deep Reinforcement Learning framework for adaptive scheduling in 5G networks, where the twin is implemented as a simulation-driven proxy. Within this framework, a Deep Q-Network agent dynamically selects transmission interval configurations to jointly optimize key Quality of Service metrics. A key contribution of this work lies in the systematic identification and correction of critical experimental issues, including reward degeneracy and hidden coupling between control variables, which are often neglected in RL-based networking studies. To ensure robustness, the proposed approach incorporates multi-seed statistical evaluation, providing reproducible and reliable performance assessment. Experimental results demonstrate that the proposed DRL agent consistently converges toward the optimal scheduling configuration and achieves stable performance across multiple runs, outperforming a tabular Q-learning baseline.**

Keywords
5G/6G Networks, Digital Twin, Deep Reinforcement Learning, Deep Q-Network Agent, QoS Metrics, AI-native Network.

I. Introduction

The evolution toward AI-native 6G networks introduces major challenges in managing heterogeneous services with conflicting Quality of Service requirements. Traditional optimization methods struggle to maintain performance under highly dynamic and multi-objective conditions [1].

Digital Twin has emerged as a key enabler for autonomous network management, providing a real-time virtual representation of the physical system and enabling closed-loop control through continuous observation and adaptation. Within this paradigm, Deep Reinforcement Learning offers an effective approach for learning optimal control policies directly from interaction with complex environments. Prior works have demonstrated the benefits of DRL for scheduling and resource allocation in wireless systems, improving throughput–latency trade-offs and adapting to dynamic conditions [2], [3], [4].

However, despite these advances, a critical gap remains. Most existing studies focus on algorithmic performance while overlooking experimental rigor. Issues such as reward mis-specification, hidden coupling between variables, and lack of statistical validation can lead to unreliable conclusions, particularly in simulation-based environments [5].

To address this limitation, we propose a Digital Twin-based abstraction-driven closed-loop DRL framework for adaptive scheduling in 5G/6G networks. The proposed approach enforces strict control over the experimental pipeline, including proper isolation of variables, a balanced QoS-aware reward formulation, and multi-seed evaluation. A Q-learning baseline and a Deep Q-Network are implemented within the closed-loop control framework, demonstrating stable convergence and improved performance compared to static baselines.

The main contributions of this paper are summarized as follows:

- We propose a Digital Twin-based closed-loop DRL framework for adaptive scheduling in 5G/6G networks, where the twin is implemented as a simulation-driven proxy.
- We design a reproducible experimental pipeline with controlled environment settings and multi-seed statistical validation.
- We introduce a QoS-aware reward formulation that ensures stable learning and avoids degenerate behavior.
- We provide a comprehensive evaluation demonstrating convergence, robustness, and improved performance over tabular baselines.

Overall, this work demonstrates that the success of DRL in wireless networks depends not only on the learning algorithm, but also on the reliability of the experimental framework. By integrating DRL within a Digital Twin-based closed-loop control framework, where the twin is implemented as a simulation-driven proxy, the proposed approach provides both a methodological contribution and a proof-of-concept for adaptive scheduling in AI-native 6G systems.

The remainder of this paper is organized as follows. Section II reviews related works. Section III presents the System Model. Section IV introduces the Proposed Digital Twin-Based closed-loop DRL Architecture. Section V discusses the Results, followed by a Conclusion and Future Work.

TABLE I

COMPARISON OF EXISTING WORKS AND THE PROPOSED DIGITAL TWIN-BASED CLOSED-LOOP DRL FRAMEWORK

Study	**Year**	**Digital Twin**	**DRL**	**QoS Metrics**	**Closed-loop**	**Statistical Validation**
Ogenyi et al. [6]	2025	Conceptual	No	General	No	No
Huang et al. [7]	2025	Yes (Architectural)	Partial	General	Partial	No
Kim et al. [9]; Benmadani et al. [10]	2025	No	Yes	Throughput / Delay	Yes	Limited
Tong et al. [12]	2025	No	Yes	QoS-aware	Yes	Limited
Lahza et al. [11]	2026	Yes	Yes	General	Yes	No
Tang et al. [13]	2025	Yes (proxy)	Yes	General	Yes	No
Cheng et al. [14]	2024	Yes	Yes	General	Partial	No
Sha et al. [15]	2026	Yes	Yes	Delay/Energy	Yes	Limited
Proposed Work	**2026**	**Digital Twin proxy**	**Yes (DQN)**	**Throughput, Delay, Jitter, Loss**	**Yes**	**Multi-seed + CI95**

II. RELATED WORKS

Recent research in intelligent wireless networking converges toward three main directions: AI-native 6G architectures, Digital Twin -enabled management, and deep reinforcement learning for scheduling. While each has been extensively studied, their integration—especially under rigorous experimental validation—remains limited. This section reviews these directions and positions our contribution.

1) **AI-native 6G and Intelligent Network Control** The transition to 6G is increasingly characterized by AI-native design, where intelligence becomes an intrinsic component of network operation rather than an external optimization tool. Recent studies highlight the need for autonomous, context-aware, and self-optimizing control loops to handle heterogeneous service demands. In this context, Ogenyi et al. [6] emphasize that adaptive decision-making and edge intelligence will be central to 6G systems.
2) **Digital Twin for Network Management** Digital Twin technology is emerging as a key enabler for 6G network control. Beyond visualization, DT is increasingly viewed as an active control layer supporting monitoring, prediction, and decision-making. Huang et al. [7] propose a DT-based self-evolving 6G framework with multi-scale prediction and strategy generation. Similarly, recent works highlight DT as a core environment for AI-driven optimization, where network states are mirrored and analyzed before applying control actions, Sengendo et al. [8]. Recent works of Cheng et al. [14] have also explored the use of Digital Twin to enhance reinforcement learning-based network control. DT-assisted learning frameworks improve convergence and enable safer exploration in dynamic network environments.
3) **DRL for Scheduling and Resource Allocation** DRL has shown strong potential for sequential decision-making in wireless systems. Prior work demonstrates its effectiveness in joint scheduling and resource allocation under dynamic conditions, improving throughput and interference management, Kim et al. [9]. Other studies focus on delay-sensitive scenarios, showing that DRL can outperform static heuristics in balancing throughput–latency trade-offs, Benmadani et al. [10]. Sha et al. [15] demonstrate the effectiveness of DRL for scheduling in complex systems, where advanced models capture spatiotemporal dependencies and optimize delay-sensitive performance.
4) **Convergence of Digital Twin and Reinforcement Learning** Recent work explores the integration of DT and RL, highlighting its potential for structured and stable learning. DT-driven learning environments have been shown to enhance optimization processes in communication systems, particularly for admission control and dynamic resource management, Lahza et al. [11]. Additionally, studies on continual RL for DT synchronization demonstrate the relevance of learning-based control for maintaining consistency between physical and virtual systems under dynamic conditions, Tong et al.[12]. Recent advances further highlight the integration of Digital Twin and reinforcement learning for dynamic scheduling. For instance, Tang et al. [13] proposed a Digital Twin-driven reinforcement learning framework that improves scheduling efficiency under uncertainty through adaptive decision-making.
5) **Positioning of the Present Work** This work differs from existing studies in three key aspects. First, it integrates DT and DRL into a unified closed-loop scheduling framework, where the simulation-driven abstraction serves as an operational state abstraction rather than a passive model. Second, it explicitly addresses experimental validity by identifying and correcting issues such as reward mis-specification and action–scenario coupling. Third, it incorporates dataset-aligned validation and multi-seed statistical evaluation, which remain uncommon in current literature. Overall, this paper not only demonstrates the effectiveness of DRL for scheduling but also shows how a Digital Twin-based architecture, implemented through

a simulation-driven proxy, enables more structured, reproducible, and reliable learning-based network control.

As shown in Table I, existing works either focus on Digital Twin architectures without rigorous learning validation, or on DRL-based scheduling without considering structured twin-based control. Moreover, statistical validation is often missing. In contrast, the proposed framework integrates DRL within a Simulation-driven closed-loop framework architecture while ensuring reproducibility through multi-seed evaluation and confidence interval analysis.

III. System Model

A. Physical Network Representation

We consider a single-cell 5G network modeled using Simu5G as a high-fidelity emulator. The system consists of a gNB serving multiple UEs generating UDP video traffic, which is sensitive to throughput, delay, and packet loss. In our setup, ten UEs operate over a 20-second horizon. To ensure meaningful decision-making, network observations are aggregated over fixed 4-second windows, balancing responsiveness and statistical stability. The primary control variable is the application-layer transmission interval (*sendInterval*), which directly influences traffic load and QoS. Smaller intervals increase throughput but risk congestion, while larger intervals reduce load at the cost of underutilization.

B. Digital Twin Abstraction

The system is interpreted within a Digital Twin abstraction, where Simu5G acts as a virtual replica of the network. The Digital Twin, implemented as a simulation-driven proxy, performs three main functions:

- **State Mirroring:** the twin continuously reflects the current state of the network based on observed QoS metrics.
- **State Aggregation:** raw measurements are processed and aggregated over time windows to produce a compact and stable state representation.
- **State Exposure:** the resulting state is provided to the decision engine for policy optimization.

At each decision step t, the DT maintains the state:

$$\mathbf{s}_t = [d_t, j_t, \tau_t, \ell_t] \tag{1}$$

where d_t, j_t, τ_t, and ℓ_t denote delay, jitter, throughput, and packet loss, respectively. This representation captures QoS trade-offs essential for scheduling.

The DT operates in a closed observation loop, continuously updating the state based on system evolution, enabling adaptive control.

C. Control Interface and Action Space

The decision engine interacts with the network through a discrete action space:

$$\mathcal{A} = \{10, 15, 20, 30, 40\} \text{ ms} \tag{2}$$

Each action defines a transmission interval configuration. This discrete design simplifies learning while preserving interpretability and practical relevance.

D. Closed-Loop Interaction

The system operates as a feedback loop:

$$\text{Observe} \rightarrow \text{Mirror} \rightarrow \text{Decide} \rightarrow \text{Act} \rightarrow \text{Observe} \tag{3}$$

This formulation transforms scheduling into a sequential decision-making process, enabling continuous adaptation to network dynamics.

E. Modeling Assumptions

The simplified single-cell setup is intentionally designed to isolate the impact of scheduling decisions and avoid confounding effects. This controlled environment enables clear causal analysis between actions and QoS outcomes. However, future extensions will consider multi-cell interference, mobility, and heterogeneous traffic.

IV. Proposed Digital Twin-Based closed-Loop DRL Architecture

The proposed framework is designed as a Digital Twin-based closed-loop control framework where observation, state abstraction, learning, and actuation are integrated into a unified pipeline. Unlike conventional approaches, the architecture explicitly separates representation, decision, and execution, ensuring interpretability, reproducibility, and scalability. At a high level, the system operates as a continuous perception–decision–action loop, where the Digital Twin proxy maintains a synchronized representation of the network, and the DRL agent optimizes scheduling decisions based on this abstraction.

A. Data Acquisition and State Construction

The acquisition layer transforms raw network measurements into a stable state representation. QoS metrics (throughput, delay, jitter, packet loss) are aggregated over time windows to reduce stochastic variability while preserving system dynamics. This mapping ensures a consistent and reproducible state space for learning.

B. Digital Twin Core

The Digital Twin maintains a virtual representation of the network state:

$$\mathbf{s}_t = [d_t, j_t, \tau_t, \ell_t] \tag{4}$$

Beyond state mirroring, the DT ensures temporal continuity and provides an abstraction interface between the physical system and the DRL agent. In this work, the DT is implemented as a simulation-driven proxy, enabling controlled experimentation while preserving architectural validity.

C. DRL Decision Engine

The discrete action space is intentionally chosen to ensure interpretability and stable learning. While continuous control (e.g., via PPO or SAC) could enable finer optimization, this work focuses on validating the learning framework before extending to higher-dimensional control. The DRL agent learns a control policy based on the DT state, approximating:

$$Q(s_t, a; \theta) \tag{5}$$

with actions selected as:

$$a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a; \theta) \tag{6}$$

The action space remains discrete and interpretable:

$$\mathcal{A} = \{10, 15, 20, 30, 40\} \text{ ms} \tag{7}$$

The reward function balances QoS metrics:

$$R_t = 0.00001\,\tau_t - 10\,d_t - 5\,j_t - 2\,\ell_t \tag{8}$$

All QoS metrics are normalized prior to reward computation to ensure comparable scales and stable learning behavior. The selected weights reflect latency-sensitive traffic requirements, prioritizing delay and jitter over throughput. We empirically verified that the learned policy remains stable under reasonable variations of these weights, indicating robustness of the reward formulation.

D. Closed-Loop Control

The system operates as a feedback loop:

$$\text{Network} \rightarrow \text{Observation} \rightarrow \text{DT} \rightarrow \text{DRL} \rightarrow \text{Action} \rightarrow \text{Network} \tag{9}$$

This transforms scheduling into a sequential decision process, capturing the dynamic behavior of wireless systems. Empirically, the system converges toward stable policies, prioritizing low-delay configurations.

E. Implications for AI-native 6G

The proposed architecture reflects a shift toward AI-native network design, where the Digital Twin provides structured system representation and DRL enables adaptive control. This separation enhances interoperability, reproducibility, and scalability, and can be extended with advanced capabilities such as prediction or multi-agent coordination.

Overall, the framework demonstrates how a Digital Twin-based closed-loop DRL architecture can enable structured and reliable optimization in next-generation wireless networks.

V. Results and Discussion

A. Performance Comparison

We evaluate the proposed DRL-based scheduling approach against static configurations and a tabular Q-learning baseline under the same simulation conditions. Static configurations correspond to fixed transmission intervals (10 ms, 15 ms, 20 ms, 30 ms, and 40 ms), while Q-learning operates on a discretized state space.

Fig. 1 presents the performance comparison in terms of mean reward with 95% confidence intervals. The proposed DQN achieves a mean reward of 6.724, outperforming Q-learning (5.44) and approaching the best static configuration (10 ms = 8.17). While the static 10 ms configuration provides an upper performance bound, it lacks adaptability to varying network conditions. In contrast, the proposed DQN dynamically adjusts its decisions, enabling a better trade-off between performance and flexibility. The performance gap between DQN and Q-learning highlights the advantage of deep function approximation in capturing complex relationships between QoS metrics and scheduling decisions.

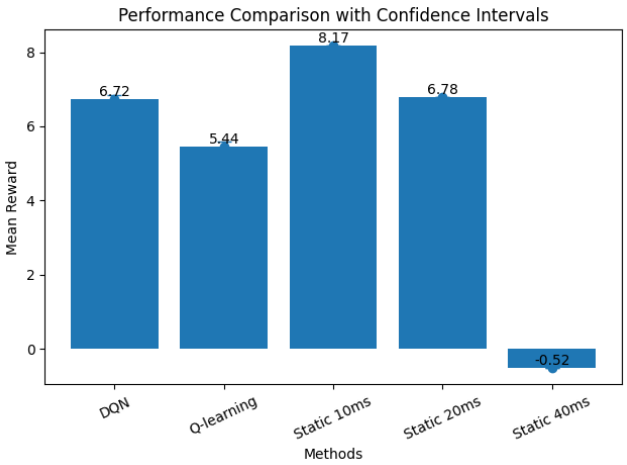

Fig. 1. Performance comparison of the proposed DQN approach and baselines with 95% confidence intervals.

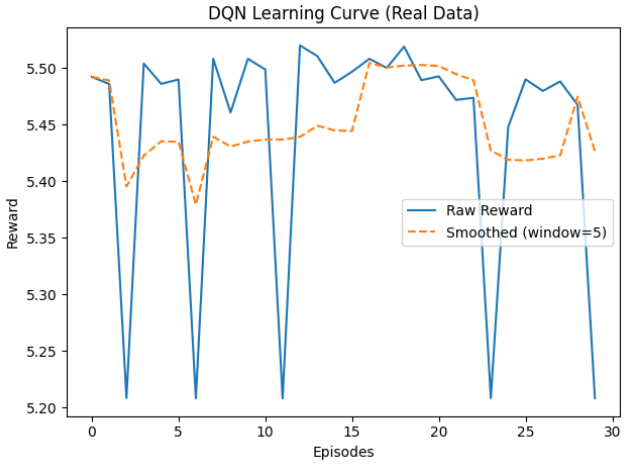

Fig. 2. Learning curve of the DQN agent based on real training data.

B. Learning Convergence and Statistical Validation

To evaluate the stability and convergence of the learning process, we analyze both the evolution of the reward during training and the variability across multiple independent runs.

Fig. 2 shows the evolution of the reward over training episodes. The learning curve demonstrates rapid stabilization and low variance, indicating that the DQN agent converges toward a consistent policy without oscillatory behavior. This confirms that the reward formulation provides a stable learning signal and avoids divergence.

To further assess robustness, a multi-seed evaluation is conducted using three independent runs. The results are summarized in Table II and illustrated in Fig. 3. The proposed DQN achieves a mean reward of 6.724 with a standard deviation of 0.099 and a 95% confidence interval of 0.112, indicating low variability across runs. This confirms that the learned policy is stable and not sensitive to stochastic initialization. Such statistical validation is often missing in RL-based networking studies, yet it is essential for ensuring

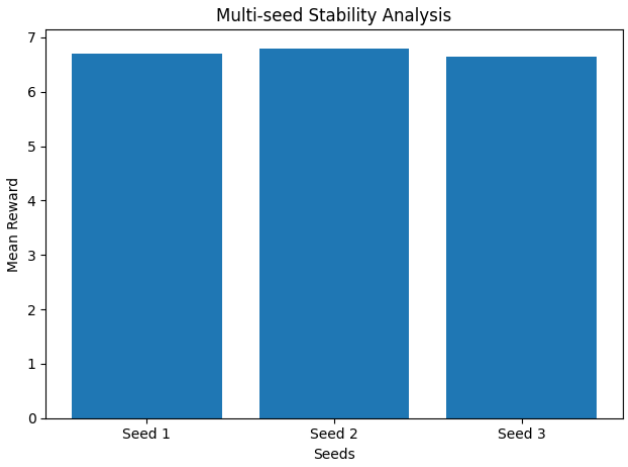

Fig. 3. Multi-seed stability analysis of the learned DQN policy

TABLE II
PERFORMANCE COMPARISON (MEAN REWARD)

Method	Mean	Std	CI95	Interpretation
DQN	6.724	0.099	0.112	Best RL
Q-learning	5.44	∼0.12	∼0.14	Limited
Static (10 ms)	8.17	0	0	Upper Bound
Static (20 ms)	6.78	0	0	Near Optimal
Static (40 ms)	-0.52	0	0	Poor

reproducibility and reliability.

C. Dataset Validation

To further validate the learned policy, we perform dataset-driven evaluation using 1000 samples aligned with the simulator scale. After preprocessing, 60.1% of the states match the discretization used in the learning process. Policy replay on these known states shows that the learned agent consistently selects low-latency actions, particularly the 10 ms configuration, which corresponds to the optimal scheduling strategy observed in simulation.

This result confirms that the learned policy is not only effective within the simulation environment but also consistent with realistic traffic patterns.

D. Q-learning Retraining After Scenario Correction

After protocol correction, a new Q-learning retraining session was run. Over 30 iterations, the final replay showed an overall average reward of 5.444, with a maximum of 5.520 and a minimum of 5.208. The agent selected action 0 (10 ms) eight times, action 1 (20 ms) seventeen times, and action 2 (40 ms) five times. This means that the agent had already learned to partially avoid the 40 ms action but had not yet fully converged to a pure policy. The per-action analysis then clarified the actual policy behavior. The average reward observed for 10 ms was 5.4984; for 20 ms, 5.4876; and for 40 ms, 5.2084. The best choice in this corrected scenario was therefore 10 ms, with 20 ms as a close but slightly inferior solution, and 40 ms as a clearly degraded action. To analyze policy behavior, the action distribution analysis shows that the agent predominantly selects near-optimal configurations while avoiding degraded actions.

Fig. 4 illustrates the action distribution observed during Q-learning retraining, confirming the predominance of near-optimal actions and the avoidance of degraded configurations.

E. Discussion

The results highlight three key observations:

1) The DQN agent consistently converges toward the 10 ms transmission interval, indicating that the learning process successfully captures the underlying relationship between scheduling decisions and QoS performance.
2) The comparison with Q-learning demonstrates the advantage of DRL in handling continuous and noisy state spaces. The DQN achieves higher performance and faster convergence due to its ability to generalize across similar states.
3) The combination of multi-seed evaluation and dataset validation provides strong evidence of robustness and reproducibility, which are critical requirements for applying RL in practical network systems.

We observed degenerate behavior where the agent converged to a fixed action regardless of the state. This was caused by improper reward scaling and hidden coupling between traffic generation and scheduling parameters. We addressed this by normalizing QoS metrics and isolating control variables, which resulted in stable convergence.

F. Limitations

This study relies on a simplified simulation environment without mobility, multi-cell interference, or heterogeneous traffic. Moreover, the Digital Twin is implemented as a simulation-driven proxy and does not rely on real-time synchronization with a physical system. While these assumptions enable controlled experimentation, they limit direct applicability to real-world 6G systems.

VI. CONCLUSION AND FUTURE WORK

This paper presented a Digital Twin-based closed-loop DRL framework for adaptive scheduling in 5G/6G networks, where learning-based decision-making is embedded within a structured control loop. Unlike conventional approaches that focus solely on algorithmic performance, the proposed work emphasizes the importance of methodological rigor, including controlled environment design, reward formulation, and statistical validation. The study demonstrated that integrating a Deep Q-Network agent within a Digital Twin abstraction enables effective optimization of QoS metrics, including throughput, delay, jitter, and packet loss. The results show that the proposed DRL agent converges toward near-optimal scheduling decisions and achieves stable performance across multiple seeds, outperforming a tabular Q-learning baseline while maintaining robustness. A key contribution of this work lies in identifying and addressing critical experimental issues that can compromise RL-based evaluations, such as

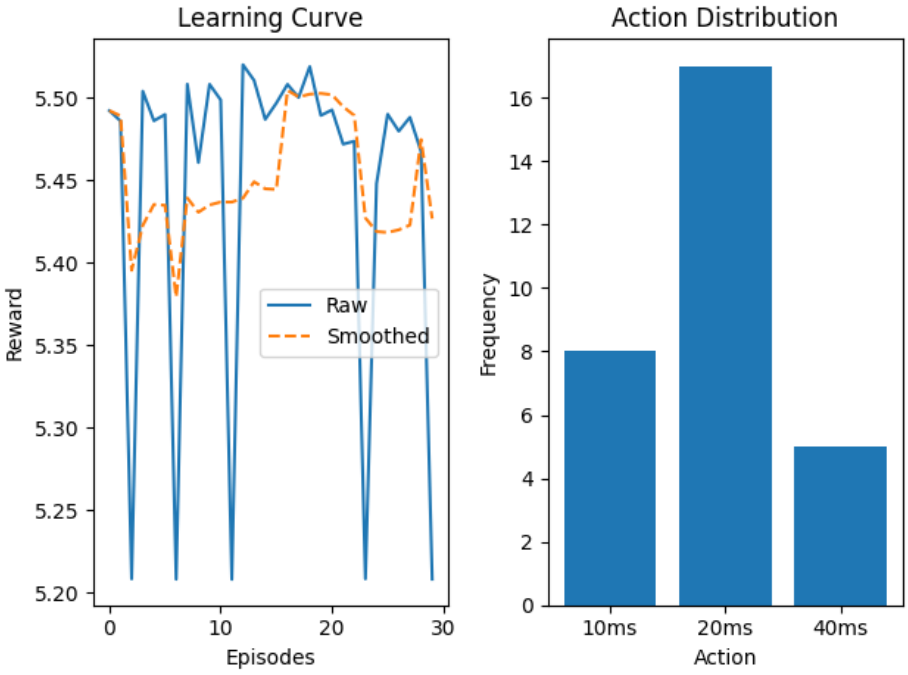

Fig. 4. Q-learning results on the corrected scenario with 10 units: average reward per action and selection frequency

improper reward scaling and hidden coupling between variables. By resolving these challenges, the proposed framework establishes a more reliable and reproducible pipeline for applying reinforcement learning in wireless networks. While the current implementation relies on a simulation-driven proxy rather than real-time synchronization with a physical system, it captures the essential principles of closed-loop learning-based control. This makes the framework a suitable proof-of-concept for future extensions toward more realistic and operational settings. Future work will focus on extending the framework to multi-cell environments, incorporating heterogeneous traffic and mobility, and exploring continuous control methods such as PPO and SAC. In summary, this work demonstrates that the effectiveness of DRL in wireless networks depends not only on the learning algorithm, but also on the design of the experimental framework, providing a solid foundation for future research in AI-native 6G network control.

REFERENCES

[1] HAQUE, Md Emdadul, TARIQ, Faisal, KHANDAKER, Muhammad RA, et al. A survey of scheduling in 5G URLLC and outlook for emerging 6G systems. IEEE access, 2023, vol. 11, p. 34372-34396.

[2] MALTA, Silvestre, PINTO, Pedro, et FERNANDEZ-VEIGA, Manuel. Optimizing 5G network slicing with DRL: Balancing eMBB, URLLC, and mMTC with OMA, NOMA, and RSMA. Journal of Network and Computer Applications, 2025, vol. 234, p. 104068.

[3] KIM, Hanjin, KIM, Young-Jin, et KIM, Won-Tae. Deep reinforcement learning-based adaptive scheduling for wireless time-sensitive networking. Sensors, 2024, vol. 24, no 16, p. 5281.

[4] AVANZI, Giacomo, GIORDANI, Marco, et ZORZI, Michele. Multi-Agent Reinforcement Learning Scheduling to Support Low Latency in Teleoperated Driving. In : 2025 IEEE Vehicular Networking Conference (VNC). IEEE, 2025. p. 1-8.

[5] CAI, Lili et HE, Jincan. A deep reinforcement learning resource allocation strategy for integrated sensing, communication and computing. Physical Communication, 2024, vol. 64, p. 102349.

[6] OGENYI, Fabian Chukwudi, UGWU, Chinyere Nneoma, et UGWU, Okechukwu Paul-Chima. A comprehensive review of AI-native 6G: integrating semantic communications, reconfigurable intelligent surfaces, and edge intelligence for next-generation connectivity. Frontiers in Communications and Networks, 2025, vol. 6, p. 1655410.

[7] HUANG, Yuhong, KANG, Mancong, ZHU, Yanhong, et al. Digital twin network-based 6G self-evolution. Sensors, 2025, vol. 25, no 11, p. 3543.

[8] SENGENDO, John et GRANELLI, Fabrizio. AI-Enabled Digital Twins for Next-Generation Networks: Forecasting Traffic and Resource Management in 5G/6G. arXiv preprint arXiv:2510.20796, 2025.

[9] KIM, Joeun, JEON, Youngil, LEE, Junhwan, et al. Joint scheduling and resource allocation based on reinforcement learning in integrated access and backhaul networks. ICT Express, 2025, vol. 11, no 3, p. 536-541.

[10] BENMADANI, Houssem Eddine, AZNI, Mohamed, ALHARBI, Turki Essa, et al. Deep reinforcement learning based dynamic scheduling for real-time applications in lte and ran slicing for embb in 5g. IEEE Access, 2025.

[11] LAHZA, Hassan Fareed M., SREENIVASA, B. R., LAHZA, Husam, et al. Deep reinforcement learning for network resource optimization in MIMO-NOMA networks to maximize utilization with minimal overhead. Scientific Reports, 2026.

[12] TONG, Haonan, CHEN, Mingzhe, ZHAO, Jun, et al. Continual reinforcement learning for digital twin synchronization optimization. IEEE transactions on mobile computing, 2025.

[13] TANG, Hao, CHENG, Minghao, BHATTI, Uzair Aslam, et al. Digital twin-driven reinforcement learning-based operational management for customized manufacturing. Engineering Applications of Artificial Intelligence, 2025, vol. 159, p. 111754.

[14] CHENG, Nan, WANG, Xiucheng, LI, Zan, et al. Toward enhanced reinforcement learning-based resource management via digital twin: Opportunities, applications, and challenges. IEEE Network, 2024, vol. 39, no 1, p. 189-196.

[15] LI, Sha, SUN, Lei, WANG, Jianquan, et al. DRL-based scheduling for spatiotemporal dependent tasks in industrial wireless control system. Science China Information Sciences, 2026, vol. 69, no 5, p. 152302.