Computer Science and Information Systems 23(1):299–320 https://doi.org/10.2298/CSIS251015009H

Robust QoS-Aware Network Scheduling for Smart Substations via Multi-Agent Adversarial Reinforcement Learning

Ping He, Dongsheng Jing, Baozhen Qi, Yu Yang, and Jingsong Xue

State Grid Suzhou Power Supply Company
Suzhou Jiangsu 215004, China
{hep_sz, jds_sz, qibz, yang_yu, js.xue }@js.sgcc.com.cn

Abstract. With the rapid development of modern power systems, traditional scheduling and reinforcement learning methods often fail to meet stringent Quality of Service (QoS) demands for low latency, high reliability, and stable bandwidth under large-scale bursty traffic. To address this problem, we propose a QoS-driven routing optimization approach based on Adversarial Reinforcement Learning, referred to as Adversarial Critic-Cooperative Actor (ACCA). By introducing adversarial agents that model worst-case perturbations, ACCA establishes a multi-agent game framework that enhances policy robustness and adaptability in dynamic network environments. Furthermore, a multi-dimensional state representation and a QoS-aware cost function are designed to capture metrics such as delay, bandwidth utilization, queue length, and packet loss. Experiments demonstrate that ACCA outperforms traditional routing protocols and standard reinforcement learning algorithms in terms of end-to-end delay, load balancing, and throughput, thereby providing an effective solution for QoS assurance in intelligent power communication networks.

Keywords: Bandwidth Management, Deep Reinforcement Learning, Adversarial Learning, Quality of Service.

1. Introduction

As the core hub of the current power system, smart substations not only perform critical functions such as data acquisition, processing, and intelligent control but also achieve deep integration among primary equipment, protection devices, and communication networks through the deployment of intelligent electronic devices[1,2]. Smart substations can support predictive maintenance by leveraging artificial intelligence, thereby enhancing operational efficiency and grid resilience.

However, smart substations impose stringent communication requirements. Remote control is the most critical service, as it directly affects grid safety and stability. In contrast, auxiliary services, such as real-time monitoring, state awareness, video transmission, and online analysis, also demand low latency, high reliability, and stable bandwidth [3,4,5]. In practice, power communication networks face substantial and bursty traffic, especially during fault detection or control command dispatch, and wireless links at distribution and sensing layers further limit bandwidth and stability [6,7]. Effectively managing high-volume, bursty, and wireless-constrained traffic is therefore critical for safe and stable grid operation. Conventional scheduling strategies, such as Open Shortest Path

First (OSPF), rely on static link costs and lack global state awareness, making them unable to adapt to topology changes, resource contention, or sudden congestion [8,9]. Reinforcement Learning (RL) has recently been explored for network scheduling, showing promise in improving adaptability and resource utilization. For example, in Deterministic Networking (DetNet), A3C-MSO frameworks leverage Graph Convolutional Networks (GCNs) to learn multiple routing and scheduling policies with improved generalization [10,11,12]. However, most RL-based approaches still overlook disturbances such as link failures, traffic surges, and environmental noise [13], which limit the robustness of policies in industrial networks.

To address these limitations, we propose a novel QoS-driven routing optimization approach based on Adversarial Reinforcement Learning [14], referred to as Adversarial Critic-Cooperative Actor (ACCA). The ACCA approach is specifically designed for heterogeneous industrial networks under the intelligent control paradigm [15]. It integrates global network state perception, intelligent routing, and disturbance modeling into a unified training framework, aiming to enhance the policy's generalization ability and robustness under quality of service constraints through adversarial interactions. Specifically, ACCA consists of a target actor agent and two adversarial agents. These adversaries simulate realistic disturbances in task conditions (e.g., bandwidth fluctuations, latency jitter) and cost conditions (e.g., malicious link injections, packet drops). Through adversarial training, the target policy learns to cope with dynamic and hostile conditions, leading to enhanced robustness and adaptability. For state modeling, the intelligent control plane is abstracted as a graph structure. Protocols like the Link Layer Discovery Protocol (LLDP) and the OpenFlow control interface (OpenFlow) are employed to collect multidimensional metrics—such as delay, bandwidth utilization, queue length, and packet loss—to construct a high-dimensional state space suitable for deep reinforcement learning [16,17]. Disturbance modeling is achieved via adjustable perturbation coefficients, enabling controlled simulation of task-level and cost-level noise. During path optimization, a QoS-driven cost function and reward design are employed to ensure a balance between performance and safety.

The main contributions of this study are summarized as follows:

- We propose ACCA, a robust QoS optimization framework based on adversarial reinforcement learning, tailored explicitly for intelligent control–based industrial networks, which significantly enhances the reliability of traffic scheduling under complex dynamics.
- We design a multi-agent adversarial training architecture that models both task and cost disturbances to enhance policy robustness against uncertain environmental factors.
- We construct a multi-dimensional network state representation and a QoS-aware path cost function to support fine-grained decision-making and adaptive control.
- We evaluate the proposed method on delay, load balancing, and throughput, and perform ablation studies to assess the impact of perturbation mechanisms on learning outcomes.

In summary, the ACCA algorithm introduces adversarial disturbance modeling into RL-based scheduling frameworks, leveraging centralized control and programmability of intelligent control to achieve robust and adaptive QoS optimization. This method not only enhances the intelligence and cost-effectiveness of industrial network scheduling but also provides a novel technical approach and theoretical foundation for optimizing communication systems in intelligent manufacturing.

The remainder of this paper is organized as follows: Section 2 reviews existing approaches for QoS routing optimization and robustness enhancement. Section 3 introduces the proposed framework, which includes reinforcement learning preliminaries, a formal problem definition, and an adversarial multi-agent architecture. Section 4 presents the QoS optimization architecture built upon Adversarial Reinforcement Learning and formulates the corresponding optimization model. Section 5 describes the experimental design and evaluation metrics. Section 6 reports and analyzes the results. Section 7 concludes the paper and outlines future research directions.

2. Related Work

To comprehensively understand the research progress in QoS routing optimization and robustness enhancement, this section provides a structured review from three perspectives: QoS Routing Based on SPF and Heuristic Enhancements, Reinforcement Learning for Adaptive QoS Routing, and Adversarial Learning for Robust Routing Policy Optimization.

2.1. QoS Routing Based on SPF and Heuristic Enhancements

Shortest Path First algorithms and their variants have been widely adopted in traditional IP networks. Representative protocols such as the Open Shortest Path First and Routing Information Protocol [18] typically rely on single network metrics (e.g. , delay or bandwidth) for path selection. While these approaches perform well under simple topologies or controllable traffic conditions, they often lead to traffic concentration on specific links in complex or high-load networks, causing congestion, unbalanced resource utilization, and increased transmission delay[19].

To overcome these limitations, researchers have proposed several enhancements. For instance, Smart OSPF adapts routing decisions by sensing global network states at the destination, thereby achieving improved traffic distribution. In addition, heuristic optimization algorithms—such as Ant Colony Optimization [20], Simulated Annealing [21], and Genetic Algorithms [22] —have been introduced to improve path quality, reduce packet loss, and balance network load [23]. However, these methods often suffer from high computational complexity and low execution efficiency, making them difficult to apply in latency-sensitive industrial networks with strict QoS requirements.

2.2. Reinforcement Learning for Adaptive QoS Routing

Reinforcement learning has emerged as a promising framework for sequential decision-making problems and has been widely applied in network routing and traffic management. By continuously interacting with the environment, RL agents autonomously learn policies that adapt to dynamic network conditions and QoS constraints.

Prior work includes the algorithm proposed by Casas-Velasco et al. [24], which incorporates link state information to mitigate congestion caused by traffic aggregation. Similarly, Sendra et al. employed RL to predict underlying network behaviors and improve bandwidth scheduling under the intelligent control architectures [25]. Nevertheless, traditional Q-learning methods encounter scalability issues in large state-action spaces, as the Q-table quickly becomes prohibitively large, limiting practical deployments in large-scale networks.

To address these limitations, Deep Q-Networks [26,27] utilize deep neural networks to approximate Q-values, enabling efficient learning in large-scale topologies such as NSFNET and ARPANET [28]. However, DQN is prone to overestimating the value function, leading to sub-optimal policy decisions and reduced training stability.

2.3. Adversarial Learning for Robust Routing Policy Optimization

Adversarial Learning [29,30], initially inspired by Generative Adversarial Network (GAN) [31,32], has gained significant traction in improving model robustness and generalization. In classical GAN frameworks, a generator and a discriminator are trained in opposition: the generator aims to produce realistic samples, while the discriminator distinguishes real from fake data. This adversarial training paradigm enables models to resist external perturbations better and adapt to diverse inputs.

In recent years, adversarial mechanisms have been increasingly applied in reinforcement learning to enhance policy robustness in dynamic and potentially adversarial environments [14,33]. Specifically in routing optimization, adversarial training allows the simulation of environmental disturbances and policy deviations, encouraging agents to learn resilient and adaptive strategies [34,35]. For example, adversarially trained policies demonstrate more stable performance and enhanced QoS preservation in latency manipulation or deceptive path guidance scenarios. In network control and scheduling, Adversarial Reinforcement Learning (ARL) employs adversarial agents or discriminators to simulate disturbances and non-stationary dynamics, enabling the learning agent to develop more robust and adaptive strategies. For example, the Generative Adversarial Reinforcement Learning Scheduling (GARLSched) algorithm [36] integrates a task embedding–based discriminator to guide policy learning, thereby improving convergence stability and adaptability to unseen workloads.

3. Methodology

This section presents the proposed framework, beginning with an overview of reinforcement learning theory. We formalize the problem by defining the state space, action space, and reward function. Then, we detail the adversarial multi-agent architecture, elaborating on the learning mechanisms and interactions among the target policy agent, task-level attacker, and cost-level attacker. This framework aims to achieve robust scheduling optimization under multi-dimensional disturbances, providing theoretical support for subsequent experimental validation.

3.1. QoS-Aware Routing Optimization via Reinforcement Learning

Reinforcement learning is an interactive learning paradigm based on the Markov Decision Process (MDP)[37]. The MDP problem is typically defined as a tuple:

$$\xi = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle . \tag{1}$$

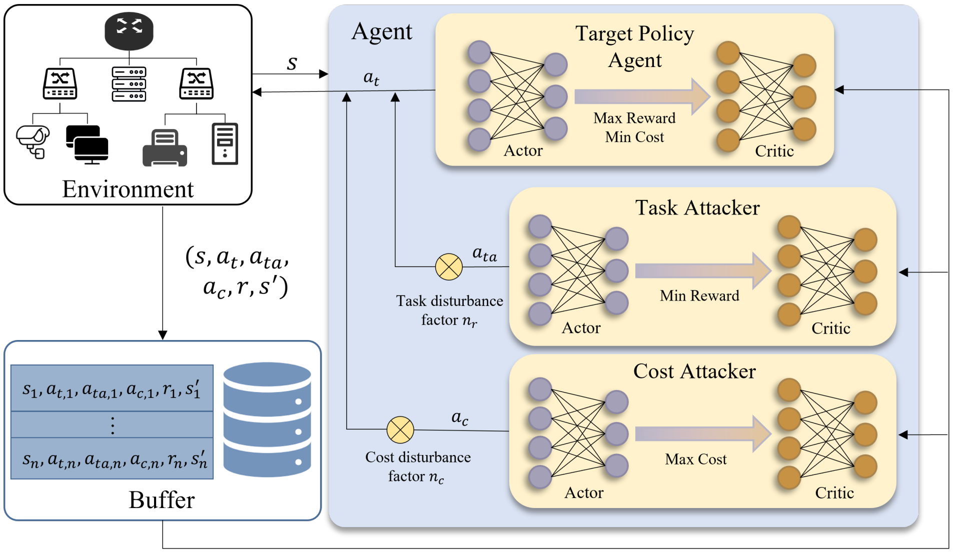

Environment
s
a_t
$(s, a_t, a_{ta}, a_c, r, s')$
Agent
Target Policy Agent
Max Reward
Min Cost
Actor
Critic
Task Attacker
a_{ta}
Task disturbance factor n_r
Min Reward
Actor
Critic
Cost Attacker
a_c
Cost disturbance factor n_c
Max Cost
Actor
Critic
$s_1, a_{t,1}, a_{ta,1}, a_{c,1}, r_1, s'_1$
⋮
$s_n, a_{t,n}, a_{ta,n}, a_{c,n}, r_n, s'_n$
Buffer

Fig. 1. Overview of the Adversarial Critic-Coordinated Actor (ACCA) architecture for secure deep reinforcement learning. The framework comprises a target policy agent and two adversarial agents (task attacker and cost attacker), all interacting with the intelligent control plane via an actor-critic learning scheme

where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to R$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor. At time t, the agent observes state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ based on policy π, transitions to s_{t+1} according to $\mathcal{P}$, and receives reward $r_t = \mathcal{R}(s_t, a_t)$. The objective is to learn an optimal policy π^* that maximizes the expected cumulative discounted reward:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{2}$$

The above Markov decision process formulation provides a general mathematical foundation for bandwidth regulation for wireless power communication networks. To apply it effectively, we need to instantiate its core components — namely, the state space, action space, and reward function — in accordance with the characteristics of data transmission and remote control tasks in smart substations.

State: In the network, the state reflects multi-dimensional performance metrics and flow scheduling information. At time t, the global state is defined as:

$$s_t = \{x_e^t \mid e \in \mathcal{E}\}, \quad x_e^t = [d_e^t, U_e^t, p_e^t, q_e^t]. \tag{3}$$

where d_e^t denotes link delay, U_e^t the bandwidth utilization, p_e^t the packet loss rate, and q_e^t the queue length at link e. These are collected by the intelligent controller via OpenFlow statistics and LLDP probes, ensuring comprehensive network observability.

Action: The action defines a selection of routing paths for multiple concurrent flow requests. Given a request d_k with candidate path set $\mathcal{P}_k = \{p_k^1, p_k^2, \ldots, p_k^K\}$, the action at time t is:

$$a_t = \{p_k \mid p_k \in \mathcal{P}_k,\ k = 1, 2, \ldots, M\} . \tag{4}$$

where M is the number of active flows. The agent learns optimal path assignments to balance QoS requirements and system robustness. For example, a video stream from host H1 to H8 may have two candidate paths $p_1^1 =$ H1–S1–S3–S6–H8, offering low latency but high load, and $p_1^2 =$ H1–S2–S5–S6–H8, offering higher stability but slightly longer delay. The agent learns to balance such trade-offs to satisfy QoS requirements while maintaining network robustness.

Reward: The reward encourages the agent to select paths that meet QoS constraints while being robust against disturbances. We define the reward as the negative total path cost under the perturbed network state $\tilde{s}_t$:

$$J(p_k, \tilde{s}_t) = \sum_{e \in p_k} \left(\lambda_1 d_e^t + \lambda_2(1 - U_e^t) + \lambda_3 p_e^t + \lambda_4 q_e^t\right) . \tag{5}$$

$$r_t = -\sum_{k=1}^{M} J(p_k, \tilde{s}_t) = -\sum_{k=1}^{M} \sum_{e \in p_k} \left(\lambda_1 d_e^t + \lambda_2(1 - U_e^t) + \lambda_3 p_e^t + \lambda_4 q_e^t\right) . \tag{6}$$

In the above formulation, $J(p_k, \tilde{s}_t)$ represents the weighted cost of flow k along the selected path p_k, incorporating s_t. The parameters λ_1–λ_4 control the relative importance of these metrics in the overall cost, balancing multiple performance objectives such as latency, bandwidth efficiency, and reliability. By taking the negative of the total path cost, the reward function encourages the agent to select routing paths with lower delay, reduced packet loss, and higher bandwidth utilization.

The actual network state under the influence of perturbations is defined as:

$$\tilde{s}_t = s_t + \eta_r \cdot \delta_t^T + \eta_c \cdot \delta_t^S . \tag{7}$$

Here, δ_t^T and δ_t^S represent task and cost perturbations, with η_r and η_c as their respective intensity coefficients. Weights λ_i reflect the relative importance of each QoS metric.

3.2. Adversarial Reinforcement Learning Architecture

To enhance the robustness of network scheduling strategies, we develop a multi-agent adversarial reinforcement learning framework. The overall framework mainly consists of three components: the environment, the agents, and the replay buffer. The agent module comprises a target policy agent and two adversarial agents — a task-level attacker and a cost-level attacker, as illustrated in Fig. 1. All agents are constructed using actor-critic architectures and are jointly deployed within the control layer to enable coordinated adversarial training:

- **Task Attacker:** Mimics performance degradation by injecting perturbations (e.g., traffic bursts), aiming to minimize the long-term reward.
- **Cost Attacker:** Simulates cost threats such as packet drops or congestion injection, aiming to maximize a long-term cost.
- **Target Policy Agent:** Learns robust routing decisions to maximize the cumulative reward under adversarial conditions.

The adversarial training process among the three agents drives the target policy to progressively adapt to dynamic perturbations, thereby ensuring resilient and stable traffic scheduling in the presence of adversarial disruptions.

Task Attacker: The task attacker is modeled using an actor-critic framework, where the actor approximates the task perturbation policy π_{task} to generate effective disturbances that degrade task performance. The critic network estimates the long-term discounted task reward function Q_r as:

$$Q_r(s_t, a_{ta}) = E_b\left[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{ta'})\right] . \tag{8}$$

where E_b denotes the empirical expectation over trajectories sampled from the experience buffer.

The critic is trained by minimizing the mean squared error (MSE) loss:

$$\mathcal{L}_{Q_r} = \text{MSE}\left(Q_r(s_t, a_{ta}), r_t + \gamma Q'_r(s_{t+1}, \pi'_{\text{task}}(s_{t+1}))\right) . \tag{9}$$

where Q'_r and π'_{task} respectively represent the target critic and target actor networks.

The actor is updated by minimizing the negative value estimated by the critic, with the loss function defined as:

$$\mathcal{L}_{\pi_{\text{task}}} = E_b\left[Q_r(s_t, \pi_{\text{task}}(s_t))\right] . \tag{10}$$

The optimal task perturbation policy is obtained by:

$$\pi^*_{\text{task}} = \arg\min_{\pi_{\text{task}}} \mathcal{L}_{\pi_{\text{task}}} . \tag{11}$$

Cost Attacker: The cost attacker also adopts an actor-critic framework, where the actor learns a perturbation policy π_{cost} aimed at maximizing long-term cost through adversarial perturbations.

The critic network estimates the discounted cumulative cost Q_c as:

$$Q_c(s_t, a_c) = E_b\left[\sum_{t'=t}^{T} \gamma^{t'-t}\, c(s_{t'}, a_{c'})\right] . \tag{12}$$

where $c(s_t, a_c)$ denotes the penalty for violating constraints at time t, and E_b is the empirical expectation over sampled trajectories.

The critic is optimized by minimizing the mean squared error (MSE) between predicted and target costs:

$$\mathcal{L}_{Q_c} = \text{MSE}\big(Q_c(s_t, a_c), \quad c_t + \gamma\, Q'_c(s_{t+1}, \pi'_{\text{cost}}(s_{t+1}))\big) . \tag{13}$$

where Q'_c and π'_{cost} are the target critic and actor networks, respectively. The actor updates its policy to maximize the expected long-term cost estimated by the critic:

$$\mathcal{L}_{\pi_{\text{cost}}} = E_b\left[Q_c(s_t, \pi_{\text{cost}}(s_t))\right] . \tag{14}$$

The optimal cost attack policy is defined as:

$$\pi^*_{\text{cost}} = \arg\max_{\pi_{\text{cost}}} \mathcal{L}_{\pi_{\text{cost}}} . \tag{15}$$

Target Policy Agent: The target policy agent also follows an actor-critic framework, where the actor learns a routing policy π_{target} to generate path selection actions, and the critic estimates the long-term discounted difference between cumulative reward and safety cost, denoted as Q_{rc}:

$$Q_{rc}(s_t, a_t) = E_b \left[\sum_{t'=t}^{T} \gamma^{t'-t} \left(r(s_{t'}, a_{t'}) - c(s_{t'}, a_{t'}) \right) \right] . \tag{16}$$

Here, $r(s_{t'}, a_{t'})$ is the immediate reward, $c(s_{t'}, a_{t'})$ is the safety cost, γ is the discount factor, and $\mathbb{E}_b[\cdot]$ denotes the expectation over the policy distribution. This evaluation function provides long-term optimization signals to the actor, enabling the agent to select robust paths that satisfy both QoS requirements and network safety in dynamic environments.

The critic network is trained by minimizing the mean squared error between predicted and target values:

$$\mathcal{L}_{Q_{rc}} = \text{MSE}\left(Q_{rc}(s_t, a_t), r_t - c_t + \gamma Q'_{rc}(s_{t+1}, \pi'_{\text{target}}(s_{t+1})) \right) . \tag{17}$$

The actor is optimized to maximize the expected utility output from the critic, using the following loss:

$$\mathcal{L}_{\pi_{\text{target}}} = -E_b \left[Q_{rc}(s_t, \pi_{\text{target}}(s_t)) \right] . \tag{18}$$

The optimal target policy is thus obtained by:

$$\pi^*_{\text{target}} = \arg\max_{\pi_{\text{target}}} E_b \left[Q_{rc}(s_t, \pi_{\text{target}}(s_t)) \right] . \tag{19}$$

Final Action Composition: In actual agent-environment interaction, the action executed by the system is the combination of the target policy and perturbations introduced by the task and cost attackers:

$$a_t = \pi_{\text{target}}(s_t) + \eta_r \cdot \pi_{\text{task}}(s_t) + \eta_c \cdot \pi_{\text{cost}}(s_t) . \tag{20}$$

Here, η_r and η_c are the scaling coefficients for the task and cost perturbations, respectively. The influence of these perturbations on system performance will be further investigated through ablation studies in subsequent sections.

Algorithm 1 Adversarial Reinforcement Learning-based QoS Optimization (ACCA)

```
**Input**: Network topology $G(V, E)$, initial QoS parameters $\theta$, convergence threshold $\epsilon$.
**Output**: Optimized routing policy $\pi^*$
Initialize actor network $\pi_\theta$ and critic network $Q_\omega$
Initialize attacker agents $\mathcal{A}_t$ (task attacker), $\mathcal{A}_s$ (cost attacker)
Initialize experience replay buffers $\mathcal{D}_\pi$, $\mathcal{D}_t$, $\mathcal{D}_s$
**for** each training episode **do**
    Initialize episode cumulative reward $R$ and cost $C$
    Initialize trajectory buffer $b$
    ************************************
    Data Collection Stage
    ************************************
    **for** each time step $t$ **do**
        Observe current network state $s_t$
        Sample actions $a_t = \pi_\theta(s_t)$, $a_{ta} = \mathcal{A}_t(s_t)$, $a_c = \mathcal{A}_c(s_t)$
        Combined action: $a_t^{total} = a_t + \eta_r a_{ta} + \eta_c a_c$
        Execute $a_t^{total}$ and observe next state $s_{t+1}$, reward $r_t$, and cost $c_t$
        Update episode totals: $R = R + r_t$, $C = C + c_t$
        Store $(s_t, a_t, a_{ta}, a_c, s_{t+1}, r_t, c_t)$ in trajectory buffer $b$
    **end for**
    Add trajectory $b$ to experience buffers $\mathcal{D}_t$, $\mathcal{D}_{ta}$, $\mathcal{D}_c$
    ************************************
    Actor-Critic Training Stage
    ************************************
    Sample mini-batches from $\mathcal{D}_t$, $\mathcal{D}_{ta}$, $\mathcal{D}_c$
    ******** Task-level Attacker Update ********
    Update critic $Q_r$ with loss $L_{Q_r}$ via Eq.8
    Update task attacker $\mathcal{A}_t$ with loss $L_{\pi_{task}}$ via Eq.10
    ******** Cost-level Attacker Update ********
    Update critic $Q_c$ with loss $L_{Q_c}$ via Eq.12
    Update cost attacker $\mathcal{A}_c$ with loss $L_{\pi_{cost}}$ via Eq.14
    ******** Target Policy Update ********
    Update combined critic $Q_{rc}$ with loss $L_{Q_{rc}}$ via Eq.17
    Update actor $\pi_\theta$ with loss $L_{\pi_{target}}$ via Eq.18
    **if** convergence criterion $|\Delta J(\pi_\theta)| < \epsilon$ is satisfied **then**
        **break**
    **end if**
**end for**
**Return** Optimized policy $\pi^* = \pi_\theta$
```

The process of learn robust scheduling policies under multi-dimensional network disturbances is shown in Algorithm 1. The algorithm consists of two main stages: Data Collection and Actor-Critic Training. During the data collection stage, the target policy agent generates actions based on the current network state, while the task-level and cost-level attacker agents introduce perturbations. The combined actions are applied to the network environment, and the resulting state transitions, rewards, and costs are recorded in trajectory buffers. These trajectories are stored in experience replay buffers for subsequent training.

In the actor-critic training stage, the task-level and cost-level attacker agents are updated to maximize adversarial objectives, enhancing the network's robustness against disturbances. Simultaneously, the target policy agent is updated using the combined critic to optimize reward and cost objectives. This iterative training process continues until the policy converges, yielding an optimized routing policy that achieves robust QoS performance under dynamic and uncertain network conditions.

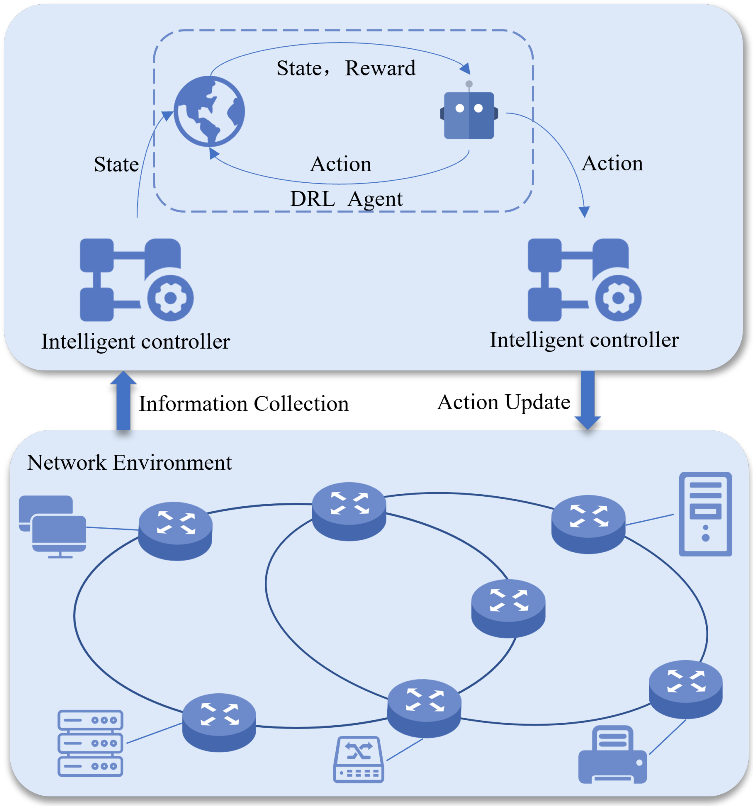


Fig. 2. The QoS optimization architecture for heterogeneous factory networks under intelligent control

4. QoS Optimization with ACCA

This section proposes a QoS optimization architecture based on Adversarial Reinforcement Learning, aiming to address the challenges of maintaining robust bandwidth control in wireless power communication networks. First, we present the architectural design and define the key network components. Then, a formal QoS optimization model is developed to provide the theoretical foundation for intelligent and resilient bandwidth scheduling under dynamic channel conditions and network disturbances.

4.1. ARL-Driven QoS Optimization Architecture

With the rapid evolution of wireless power communication networks, ensuring stable and efficient data transmission has become a fundamental requirement for modern smart grids and power systems. Diverse traffic types coexist in such networks, including real-time monitoring signals, control commands, and high-volume measurement data. These services impose heterogeneous Quality of Service requirements, where latency, bandwidth utilization, and reliability are especially critical since they directly affect grid stability and system responsiveness. However, traditional QoS-aware routing and bandwidth allocation strategies often fail to cope with the dynamic wireless environment, interference, and potential adversarial disruptions. This motivates the design of intelligent and robust optimization architectures, where adversarial reinforcement learning can provide adaptive decision-making for QoS-driven bandwidth regulation in WPCNs.

Conventional static scheduling strategies and basic reinforcement learning methods often fail to maintain robust performance in highly dynamic network states and the presence of multiple sources of disturbances (e.g., traffic bursts, link anomalies, malicious attacks). To address this, we design a multi-agent QoS optimization architecture based on ARL, as illustrated in Fig. 2, which integrates the benefits of heterogeneous factory networks, intelligent centralized control, and adversarial learning.

The architecture consists of three layers:

- **Data Plane:** Composed of OpenFlow-enabled virtual switches, responsible for efficient packet forwarding and traffic processing.
- **Control Plane:** Consists of intelligent controllers that periodically transmit Link Layer Discovery Protocol packets and use echo mechanisms to monitor network topology and link states dynamically. Port statistics are collected to extract key metrics such as bandwidth utilization, packet loss rate, and queue length.
- **Decision Plane:** Hosts ARL-based agents that leverage global network state information from the control plane to learn robust path selection and traffic scheduling strategies. This layer includes one target policy network and two adversarial networks that simulate task-level and cost-related disturbances. The adversarial outputs are weighted and injected into the network state to guide robust policy learning under complex and dynamic environments. Finally, flow table rules are generated by the intelligent controller based on the agent's outputs, enabling unified encapsulation and dynamic configuration across the data link (e.g., MAC address, VLAN ID), network (e.g., IP address), and transport layers (e.g., TCP/UDP ports), forming a flat and intelligent factory network.

4.2. QoS Optimization Modeling

In wireless power communication networks, the dynamic nature of spectrum usage and energy constraints leads to constantly evolving network states, while the QoS demands of different data flows vary significantly. Particularly in mission-critical scenarios such as industrial monitoring and autonomous unmanned systems, the network must achieve efficient and robust bandwidth allocation under constraints like interference, delay, and limited wireless resources.

Algorithm 2 QoS Optimization with ACCA

```
Require: Network graph $G(V, E)$, flow request $d_k = (s_k, t_k)$
Require: Disturbance coefficients $\eta_r, \eta_c$
Require: Learning rates $\alpha_p, \alpha_T, \alpha_C$
Ensure: Robust routing policy $\pi_\theta$
Initialize policy network $\pi_\theta$, task attacker $A_T$, cost attacker $A_C$
Initialize replay buffer $\mathcal{D}$
for each training episode do
    Observe network state $s_t$
    Adversarial disturbance generation
    $\delta_t^T \leftarrow A_T(s_t)$ via Eq.8
    $\delta_t^S \leftarrow A_C(s_t)$ via Eq.12
    $\tilde{s}_t \leftarrow s_t + \eta_r \delta_t^T + \eta_c \delta_t^S$ via Eq.23
    QoS cost evaluation
    for each path $p_k \in \mathcal{P}_k$ do
        Compute link metrics $\{d_e, U_e, p_e, q_e\}$
        Evaluate path cost $J(p_k, \tilde{s}_t)$
    end for
    Policy decision
    Sample/select path $p_k \sim \pi_\theta(\tilde{s}_t)$
    Apply forwarding rule to data plane switches
    Environment feedback
    Observe reward $r_t \leftarrow -J(p_k, \tilde{s}_t)$ and next State $s_{t+1}$ via Eq. 5
    Store transition $(s_t, \tilde{s}_t, p_k, r_t, s_{t+1})$ into $\mathcal{D}$
    Joint update
    Sample minibatch $B \subset \mathcal{D}$
    $\theta_T \leftarrow \theta_T + \alpha_T \nabla_{\theta_T} L_T(B)$
    $\theta_C \leftarrow \theta_C + \alpha_C \nabla_{\theta_C} L_C(B)$
    $\theta \leftarrow \theta - \alpha_p \nabla_\theta L_\pi(B)$
end for
Return $\pi_\theta$
```

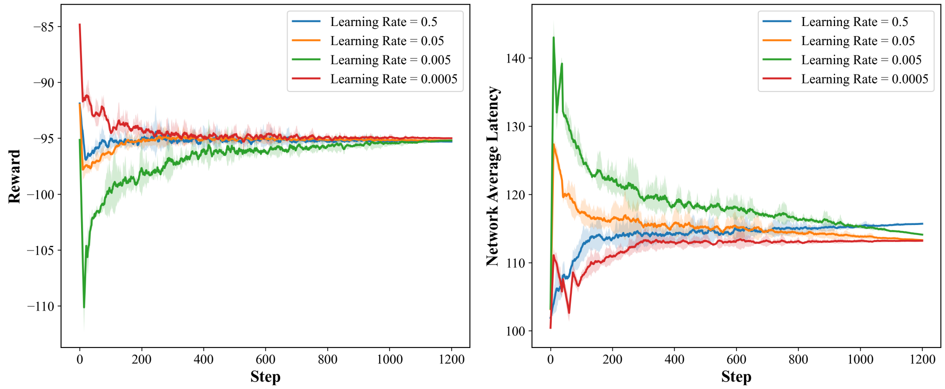
We model the wireless power communication network as an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_M\}$ denotes the set of nodes representing wireless power communication devices (e.g., access points, relays, or terminals), and $E = \{e_1, e_2, \ldots, e_N\}$ denotes the wireless links subject to fading and interference. The network controller leverages spectrum sensing and power feedback to monitor the topology and link-level states, obtaining key metrics such as signal-to-interference-plus-noise ratio, residual energy, throughput, and queue lengths.

Each link $e \in E$ at time t is described by a four-dimensional vector:

$$\mathbf{x}_e^t = [d_e^t, U_e^t, p_e^t, q_e^t] \, . \tag{21}$$

where d_e^t is the LLDP-measured delay, p_e^t is the packet loss rate, and q_e^t is the queue length as a measure of congestion. U_e^t is the bandwidth utilization calculated as:

$$U_e^{used} = \frac{\Delta \text{Rx} + \Delta \text{Tx}}{\Delta t}, \quad U_e^t = \frac{U_e^{used}}{B_e} \, . \tag{22}$$



(a) (b)

Fig. 3. Performance of the proposed algorithm under the same disturbance coefficients η_r and η_c with different learning rates α. (a) and (b) represent the reward and the average network delay, respectively

Here, B_e is the maximum bandwidth of the link, and $\varDelta$Rx, $\varDelta$Tx denote the bytes received and transmitted during interval $\varDelta t$. The overall network state at time t is via Eq. 3.

For a given flow request d_k, with source s_k and destination t_k, a set of candidate paths $\mathcal{P}_k = \{p_1, p_2, \ldots, p_K\}$ is considered. A cost function quantifies the QoS of a path via Eq. 5.

4.3. Adversarial Disturbance Modeling

In wireless power communication networks, link states are highly susceptible to various disturbances, primarily stemming from spectrum dynamics and security threats. These disturbances can be categorized into communication-level fluctuations and adversarial cost attacks. The former, denoted by the vector δ_t^T, captures uncertainties such as channel fading, spectrum contention, and traffic bursts. The latter, denoted by δ_t^S, represents malicious behaviors including jamming, interference injection, and bandwidth occupation attacks. Under such disturbances, the actual network state can be expressed as:

$$\tilde{s}_t = s_t + \eta_r \cdot \delta_t^T + \eta_c \cdot \delta_t^S \, . \tag{23}$$

where η_r and η_c are disturbance coefficients controlling the intensity of task and cost perturbations. These perturbations may increase delay, reduce bandwidth availability, increase loss rates, and lengthen queues, thus impacting routing decisions and QoS performance.

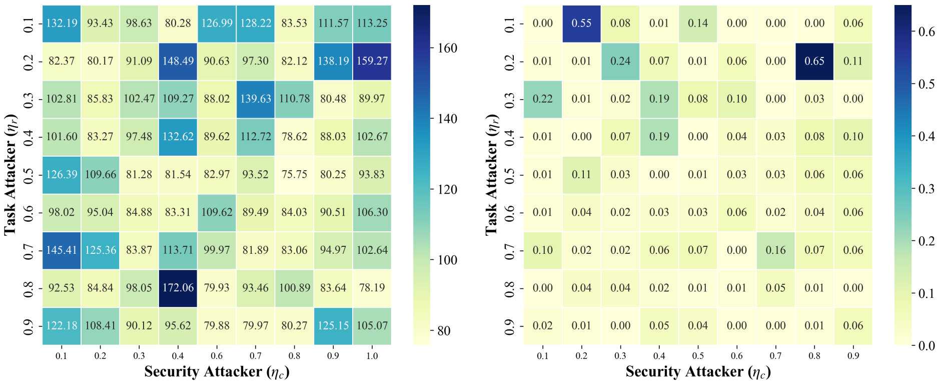
Optimization Objective: Given disturbed state modeling, the objective is to learn a robust routing policy π that minimizes the expected path cost under adversarial conditions.

$$\min_{\pi} \; E_{\delta_t} \left[J(p_k, \tilde{s}_t) \right] \, . \tag{24}$$

subject to:

- Valid probability distribution over path selection: $\sum_{k=1}^{K} w_k = 1, \quad w_k \geq 0$
- QoS constraints: $J(p_k, \tilde{s}_t) < L_{\text{max}}$
- Link capacity constraint: $U_e^t < 1, \forall e \in p_k$
- Robustness requirement: $|J(p_k, \tilde{s}_t) - J(p_k, s_t)| \leq \epsilon$

The algorithm 2 outlines the core inference procedure of ACCA, detailing the complete operational sequence from state acquisition to adversarial interaction and final path selection. This optimization framework provides the theoretical foundation for learning robust ARL-based bandwidth regulation policies, enhancing both the reliability and security of resource allocation in wireless power communication networks.



(a) Reward Heatmap

(b) Latency Heatmap

Fig. 4. Sensitivity analysis of the proposed algorithm under varying task perturbation coefficient η_r and cost perturbation coefficient η_c: (a) average episodic reward heatmap; (b) average episodic cost cost heatmap

5. Experiments

This section presents a comprehensive set of experiments designed to evaluate the effectiveness of the proposed ACCA approach. The experiments aim to analyze three aspects: the convergence behavior of the algorithm, its sensitivity to perturbation parameters, and its performance compared with baseline methods. All evaluations are conducted in a unified simulation environment using standard QoS-related metrics, including reward value, end-to-end latency, load balancing, and throughput.

5.1. Experimental Environment

To evaluate the effectiveness of the proposed method in the context of bandwidth regulation for wireless power communication networks, we developed a simulation testbed using the Mininet network emulator in combination with the Ryu intelligent controller. The simulated environment represents a smart substation scenario, where heterogeneous devices, including intelligent electronic devices, sensors, actuators, and control terminals, are interconnected. The network adopts a hybrid wired/wireless topology supporting multiple communication protocols, including IEC 61850, Modbus, Wi-Fi, and TCP/IP.

Small-scale Mininet topology: The network topology consists of 6 switches and 8 hosts, forming a hybrid wired/wireless architecture that emulates the heterogeneous communication environment of a power communication network. Among the hosts, different roles are assigned to reflect real-world scenarios, including phasor measurement units (PMUs), protection relays, intelligent electronic devices (IEDs), and remote monitoring terminals. Flow request rates range from 10 Mb/s to 120 Mb/s, supporting both low-latency monitoring signals and high-throughput control or protection data streams.

Large-scale Mininet topology: To further assess scalability and robustness, additional experiments were conducted on an extended topology with up to 20 switches and 32 hosts, introducing more complex interconnections and traffic dynamics.

This multi-scale experimental configuration enables the simulation of diverse traffic conditions under different operational states—such as routine monitoring, demand response, fault detection, and emergency control—providing a realistic and comprehensive basis for assessing bandwidth regulation and QoS optimization strategies in wireless power communication networks.

Table 1. Routing Performance Comparison on Small-scale and Large-scale Mininet Topologies at 120 Flows/s

Method	Load Imbalance(%)	Throughput (Mbps)	network latency(ms)
	Small-scale Mininet topology		
OSPF	0.75	60	116.25
DQN	0.55	63	115.25
A3C-MSO	0.50	68	114.64
ACCA	0.31	72	113.14
	Large-scale Mininet topology		
OSPF	0.83	68	131.32
DQN	0.62	73	128.79
A3C-MSO	0.57	78	124.98
ACCA	0.40	84	122.18

5.2. Experimental Setup

The proposed ARL algorithm is implemented in PyTorch. The key hyperparameters are set as follows: learning rate $\alpha = 0.001$, discount factor $\gamma = 0.95$, exploration rate $\epsilon = 0.4$, and replay buffer size of 2000. To evaluate the performance and convergence of the model, the average reward is recorded every 100 training episodes as a measure of policy improvement. Each experiment is performed with a single random seed over 1200 training steps, allowing the algorithm to interact with the simulated network environment and progressively optimize the scheduling policy.

To comprehensively evaluate the approach, three primary performance metrics are used:

- **End-to-end latency** – the average per-flow transmission delay, reflecting the efficiency of data delivery.
- **Load balancing** – measured by the standard deviation of link utilization, indicating the distribution of traffic across the network.
- **Network throughput** – the total amount of successfully transmitted data over the network, representing bandwidth utilization.

During training, the ARL agent interacts with the network environment, selects actions based on the learned policy, and receives feedback in the form of rewards. This process enables the agent to adaptively adjust routing decisions, reducing latency, balancing load, and maximizing throughput.

5.3. Baseline Methods

To evaluate the performance of the proposed ACCA algorithm, we compare it against the following baseline approaches:

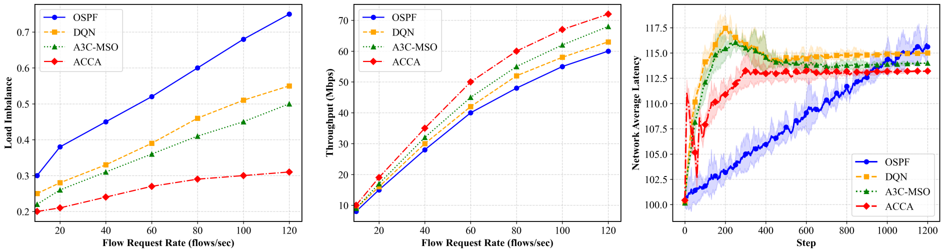
- **OSPF** – a widely used static routing protocol that determines paths based on link-state information. OSPF does not adapt dynamically to varying traffic loads, serving as a baseline for conventional routing performance.
- **DQN** – a reinforcement learning–based routing method that leverages deep Q-networks to learn optimal forwarding policies. By approximating the Q-function with neural networks, DQN can handle high-dimensional state spaces more effectively than traditional Q-learning. However, it often suffers from unstable convergence and limited generalization in large-scale or highly dynamic network scenarios, serving as a representative baseline of value-based RL approaches.
- **A3C-MSO** – an advanced RL-based QoS optimization approach designed for deterministic networking. It employs an asynchronous advantage actor–critic architecture with multipolicy learning, enabling joint routing and scheduling optimization. A3C-MSO improves modeling and generalization in complex network environments by incorporating graph convolutional networks for topology representation.

These baselines enable us to systematically evaluate the algorithm's advantages in terms of load balancing, latency reduction, and throughput maximization under dynamic network conditions.

6. Results and Analysis

6.1. Convergence and Robustness Evaluation

To evaluate the training stability and convergence capability of the proposed algorithm under perturbation environments, we first conducted a systematic assessment of the impact of different learning rates on training performance under fixed disturbance coefficients ($\eta_r = 0.5$, $\eta_c = 0.5$). Experiments were performed with a single random seed over 1200 training steps, recording the trends of average episodic reward and link latency to measure the convergence speed and stability of the model. As shown in Fig. 3, the algorithm achieves optimal performance at a learning rate of $\alpha = 0.0005$, exhibiting faster convergence, lower final network latency, and more balanced load distribution. During the initial



(a) Load Balancing (b) Throughput (c) Latency over Training

Fig. 5. Comparative performance evaluation of routing algorithms under varying traffic demand and training progress: (a) Network load imbalance versus traffic demand; (b) Network throughput versus traffic demand; (c) Average network latency during training versus training steps

training stage (approximately steps 0–50), transient congestion caused by traffic surges resulted in increased latency and decreased rewards; subsequently, the model progressively captured the network dynamics and optimized its policy. The reward curve stabilized between steps 400 and 600, with a significant reduction in latency, indicating that the policy had essentially converged and maintained stable performance thereafter.

After determining the optimal learning rate, we conducted a perturbation sensitivity study to systematically evaluate the robustness of the proposed algorithm under various combinations of the task perturbation coefficient η_r and the cost perturbation coefficient η_c, each ranging from 0.1 to 0.9. For each parameter pair, experiments were repeated three times with different random seeds, training for 500 episodes per run. Both the average episodic reward and average cost were recorded and visualized using heatmaps.

As shown in Fig. 4, the reward heatmap indicates that high-reward regions primarily cluster near the anti-diagonal of the perturbation coefficient grid, particularly when $\eta_r \leq 0.3$ or $\eta_r \geq 0.7$. This suggests that the policy performs stably and achieves superior performance under relatively weak or strong task perturbations. In contrast, moderate perturbations (e.g., $\eta_r \approx 0.5$) tend to disrupt learning, resulting in degraded policy performance. This phenomenon provides a possible explanation for the observed results: under weak perturbations, the environment is close to its nominal state, allowing the agent to learn the optimal policy with stable convergence. Under moderate perturbations, the agent receives conflicting signals—perturbations are strong enough to interfere with learning but insufficient to induce robust policy adaptation, resulting in suboptimal performance. Under strong perturbations, the agent is compelled to account for a broader range of potential environment variations during training, thereby enhancing the robustness of the learned policy and yielding higher rewards.

The cost heatmap further shows that the algorithm maintains low or near-zero average cost across most perturbation combinations, indicating robust cost defense. Elevated costs mainly occur at the boundaries $\eta_c \leq 0.4$ and $\eta_c \geq 0.8$, where the former corresponds to insufficient perturbation for effective threat recognition, and the latter reflects excessive perturbation that destabilizes training. Overall, these results highlight the nonlinear relationship between perturbation intensity and policy performance, suggesting that both very weak and extreme perturbations can be beneficial for learning stable and robust policies. In contrast, moderate perturbations may negatively affect performance.

In summary, comprehensive analysis identifies the optimal disturbance parameter configuration as task perturbation coefficient $\eta_r = 0.8$ and cost perturbation coefficient $\eta_c = 0.4$. Under this configuration, the algorithm strikes a balance between high reward acquisition and low cost, demonstrating robust and effective performance in uncertain environments.

6.2. Comparative Performance Evaluation

After tuning key hyperparameters, we benchmark the proposed method against several baseline approaches under both small-scale and large-scale network settings. As summarized in Table 1, in the large-scale topology (20 switches and 32 hosts), all methods exhibit consistent performance trends with the small-scale case (6 switches and 8 hosts). Specifically, at a flow rate of 120 flows/s, **ACCA** achieves the lowest load imbalance at 0.40%, while A3C-MSO shows a higher imbalance of 0.57%. Similarly, **ACCA** attains the highest throughput of 84 Mbps, exceeding the 78 Mbps achieved by A3C-MSO, and maintains the lowest network latency at 122.18 ms compared to 124.98 ms for A3C-MSO. Compared to the small-scale topology, absolute values such as load imbalance and latency are slightly higher, and throughput is somewhat lower due to increased network depth and path diversity. Despite this, the relative performance advantages of **ACCA** remain consistent across topologies. Therefore, for clarity, only the small-scale Mininet results are illustrated in Fig. 5.

As shown in Fig. 5(a), the degree of load imbalance increases with traffic demand across all methods. However, the growth rate differs significantly. OSPF exhibits a sharp rise in imbalance under high loads, indicating its inability to balance traffic adaptively due to its static nature. The DQN-based method alleviates imbalance more effectively than OSPF, as it leverages deep Q-learning to explore adaptive routing decisions. However, its value-based learning framework often leads to unstable convergence, and the achieved balance is still inferior to that of actor–critic approaches. The A3C-MSO method achieves a better balance than both OSPF and DQN across all flow rates, demonstrating stronger adaptive capabilities. In contrast, the algorithm consistently maintains the lowest load imbalance, even under extreme conditions (e.g., 120 flows/sec), validating its superior congestion awareness and dynamic routing capability.

Fig. 5(b) further compares the throughput performance of the four methods under increasing traffic. OSPF shows saturation in throughput at medium to high load levels, while the DQN method achieves improved bandwidth utilization but suffers from unstable performance under bursty traffic. The A3C-MSO approach outperforms both OSPF and DQN, but the proposed ACCA method delivers the highest throughput across all flow levels, with the most pronounced improvements observed in the 60–120 flows/sec range. This confirms its effectiveness in handling complex network dynamics and adversarial perturbations, maximizing resource usage, and enhancing overall QoS.

Fig. 5(c) illustrates the evolution of the average network latency during training under the four routing strategies. As a static protocol, OSPF exhibits a relatively smooth but steadily increasing delay trend throughout the training process, indicating its limited adaptability. The DQN method exhibits significant fluctuations during training, reflecting its unstable convergence behavior. Although it can reduce latency compared to OSPF, it fails to achieve consistent long-term improvements. In contrast, the A3C-MSO method shows a rapid increase in latency during the initial training stage, followed by partial stabilization in later stages; however, it still maintains a relatively high average delay. Notably, the algorithm demonstrates a brief fluctuation in the early stage, but quickly converges after approximately 400 steps and consistently maintains the lowest average latency thereafter. Throughout the entire training process, ACCA outperforms OSPF, DQN, and A3C-MSO, highlighting its superior capability in adapting to network dynamics and learning efficient routing policies, thereby effectively alleviating network congestion and enhancing data transmission efficiency.

7. Conclusion

We addressed the problem of bandwidth regulation in wireless power communication networks by proposing an adaptive routing optimization method based on Adversarial Reinforcement Learning, achieving joint optimization of Quality of Service and network cost. By incorporating the task perturbation factor η_r and the cost perturbation factor η_c, the method simulates bandwidth fluctuations and potential attacks in dynamic and uncertain network environments, thereby enhancing the agent's capability to perceive network state changes and adapt its policies accordingly. A robust training mechanism is also designed to improve the stability of policy convergence and its generalization performance.

The proposed method demonstrates significant potential in optimizing key metrics such as bandwidth allocation, link load balancing, and latency, indicating that the adversarial mechanism can effectively improve network resource utilization and enhance policy robustness. Nonetheless, the current perturbation model does not fully capture complex attacks, such as time-varying disturbances and intelligence-induced perturbations. The single-agent training framework also faces scalability limitations in multi-controller and multi-path scenarios, which are typical of wireless power communication networks.

Future work will consider integrating Graph Neural Networks or Multi-Agent Reinforcement Learning frameworks to enhance the model's representational capacity and decision-making efficiency in large-scale heterogeneous networks. Moreover, the exploration of explainability mechanisms will be pursued to improve the transparency and auditability of routing policies, providing more reliable support for practical deployment in wireless power communication networks.

Acknowledgments. This work was supported by Science and Technology Project of State Grid Jiangsu Electric Power Co., Ltd., and Science and Technology Project of Suzhou Suneng Group Co., Ltd. under Grant SGSZSNJTKJJS2500979.

References

1. Juan C Lozano, Keerthi Koneru, Neil Ortiz, and Alvaro A Cardenas. Digital substations and iec 61850: A primer. *IEEE Communications Magazine*, 61(6):28–34, 2023.
2. José Gaspar, Tiago Cruz, Chan-Tong Lam, and Paulo Simões. Smart substation communications and cybersecurity: A comprehensive survey. *IEEE communications surveys & tutorials*, 25(4):2456–2493, 2023.

3. Tze-Yang Tung and Deniz Gündüz. Deepwive: Deep-learning-aided wireless video transmission. *IEEE Journal on Selected Areas in Communications*, 40(9):2570–2583, 2022.
4. S.C. Jayasinghe, M. Mahmoodian, A. Sidiq, T.M. Nanayakkara, A. Alavi, Sam Mazaheri, F. Shahrivar, Z. Sun, and S. Setunge. Innovative digital twin with artificial neural networks for real-time monitoring of structural response: A port structure case study. *Ocean Engineering*, 312:119187, 2024.
5. Wenqing Zheng, Hao Frank Yang, Jiarui Cai, Peihao Wang, Xuan Jiang, Simon Shaolei Du, Yinhai Wang, and Zhangyang Wang. Integrating the traffic science with representation learning for city-wide network congestion prediction. *Information Fusion*, 99:101837, 2023.
6. Xiang Cheng, Dongliang Duan, Shijian Gao, and Liuqing Yang. Integrated sensing and communications (isac) for vehicular communication networks (vcn). *IEEE Internet of Things Journal*, 9(23):23441–23451, 2022.
7. Lipeng Zhu, Wenyan Ma, and Rui Zhang. Movable antennas for wireless communication: Opportunities and challenges. *IEEE Communications Magazine*, 62(6):114–120, 2023.
8. G. BalaMurugan, N. Palanivel, and A. John. Application reliable traffic control method for efficient data management in wireless-aided computer applications. In *2023 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5, 2023.
9. Quan Chen, Lei Yang, Yong Zhao, Yi Wang, Haibo Zhou, and Xiaoqian Chen. Shortest path in leo satellite constellation networks: An explicit analytic approach. *IEEE Journal on Selected Areas in Communications*, 42(5):1175–1187, 2024.
10. Wenbin Tian, Chaojie Gu, Miao Guo, Shibo He, Jiawen Kang, Dusit Niyato, and Jiming Chen. Large-scale deterministic networks: Architecture, enabling technologies, case study, and future directions. *IEEE Network*, 38(4):284–291, 2024.
11. Sijin Yang, Lei Zhuang, Jianhui Zhang, Julong Lan, and Bingkui Li. A multipolicy deep reinforcement learning approach for multiobjective joint routing and scheduling in deterministic networks. *IEEE Internet of Things Journal*, 11(10):17402–17418, 2024.
12. Ying Ma, Haijie Lou, Ming Yan, Fanghui Sun, and Guoqi Li. Spatio-temporal fusion graph convolutional network for traffic flow forecasting. *Information Fusion*, 104:102196, 2024.
13. Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *ACM Comput. Surv.*, 58(2):53:1–53:37, September 2025.
14. Maxwell Standen, Junae Kim, and Claudia Szabo. Adversarial machine learning attacks and defences in multi-agent reinforcement learning. *ACM Computing Surveys*, 57(5):1–35, 2025.
15. Rana M. Sohaib, Oluwakayode Onireti, Yusuf Sambo, Rafiq Swash, Shuja Ansari, and Muhammad A. Imran. Intelligent resource management for embb and urllc in 5g and beyond wireless networks. *IEEE Access*, 11:65205–65221, 2023.
16. Chen Wu, Shikai Guo, Hui Li, Chenchen Li, and Rong Chen. Estimating uncertainty in line-level defect prediction via perceptual borderline oversampling. *ACM Trans. Softw. Eng. Methodol.*, June 2025. Just Accepted.
17. Shuhua Deng, Zhangping Yin, and Xieping Gao. Investigating vulnerabilities in openflow discovery protocol: Novel attacks and their defense. *IEEE Transactions on Dependable and Secure Computing*, 22(6):6376–6388, 2025.
18. M.Sahaya Sheela, R. Suganthi, S. Gopalakrishnan, T. Karthikeyan, K. Jyothi, and K. Ramamoorthy. Secure routing and reliable packets transmission in manet using fast recursive transfer algorithm. *Babylonian Journal of Networking*, 2024:78–87, 06 2024.
19. Muhammad Numan, Akif Zia Khan, Mansoor Asif, Sarmad Majeed Malik, and Kashif Imran. Exploiting the inherent flexibility in transmission network for optimal scheduling, wind power utilization, and network congestion management. *IEEE Access*, 9:88746–88758, 2021.
20. Christian Blum. Ant colony optimization: A bibliometric review. *Physics of Life Reviews*, 51:87–95, 2024.

21. Yusuf Alper Kaplan. Forecasting of global solar radiation: A statistical approach using simulated annealing algorithm. *Engineering Applications of Artificial Intelligence*, 136:109034, 2024.
22. Bushra Alhijawi and Arafat Awajan. Genetic algorithms: Theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*, 17(3):1245–1256, 2024.
23. Ayodeji Olalekan Salau and Melesew Mossie Beyene. Software defined networking based network traffic classification using machine learning techniques. *Scientific Reports*, 14(1):20060, 2024.
24. Daniela M Casas-Velasco, Oscar Mauricio Caicedo Rendon, and Nelson LS da Fonseca. Intelligent routing based on reinforcement learning for software-defined networking. *IEEE Transactions on Network and Service Management*, 18(1):870–881, 2020.
25. Sandra Sendra, Albert Rego, Jaime Lloret, Jose Miguel Jimenez, and Oscar Romero. Including artificial intelligence in a routing protocol using software defined networks. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 670–674, 2017.
26. Xinqiang Chen, Ruiyang Hu, Kai Luo, Huafeng Wu, Salvatore Antonio Biancardo, Yiwen Zheng, and Jiangfeng Xian. Intelligent ship route planning via an a* search model enhanced double-deep q-network. *Ocean Engineering*, 327:120956, 2025.
27. Juan Moreno-Malo, Juan-Luis Posadas-Yagüe, Juan Carlos Cano, Carlos T Calafate, J Alberto Conejero, and Jose-Luis Poza-Lujan. Improving traffic light systems using deep q-networks. *Expert Systems with Applications*, 252:124178, 2024.
28. Yi-Ren Chen, Amir Rezapour, Wen-Guey Tzeng, and Shi-Chun Tsai. Rl-routing: An sdn routing algorithm based on deep reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 7(4):3185–3199, 2020.
29. Hao Zhang, Linfeng Tang, Xinyu Xiang, Xuhui Zuo, and Jiayi Ma. Dispel darkness for better fusion: A controllable visual enhancer based on cross-modal conditional adversarial learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26477–26486, 2024.
30. Qiankun Zuo, Huisi Wu, C. L. Philip Chen, et al. Prior-guided adversarial learning with hypergraph for predicting abnormal connections in alzheimer's disease. *IEEE Transactions on Cybernetics*, 54(6):3652–3665, 2024.
31. Xuhui Liu, Bohan Zeng, Sicheng Gao, Shanglin Li, Yutang Feng, Hong Li, Boyu Liu, Jianzhuang Liu, and Baochang Zhang. Ladiffgan: Training gans with diffusion supervision in latent spaces. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1115–1125, 2024.
32. Willone Lim, Kelvin Sheng Chek Yong, Bee Theng Lau, and Colin Choon Lin Tan. Future of generative adversarial networks (gan) for anomaly detection in network security: A review. *Computers & Security*, 139:103733, 2024.
33. Chuyao Wang and Nabil Aouf. Explainable deep adversarial reinforcement learning approach for robust autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 10(4):2551–2563, 2025.
34. Ao Qu, Yihong Tang, and Wei Ma. Adversarial attacks on deep reinforcement learning-based traffic signal control systems with colluding vehicles. *ACM Trans. Intell. Syst. Technol.*, 14(6):113:1–113:22, November 2023.
35. Juan Chafla Altamirano, Mariem Guitouni, Hassan Hassan, and Khalil Drira. Routing optimization based on drl and generative adversarial networks for sdn environments. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pages 1–5, 2024.
36. Jingbo Li, Xingjun Zhang, Jia Wei, Zeyu Ji, and Zheng Wei. Garlsched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems. *Future Generation Computer Systems*, 135:259–269, 2022.

37. Mansour Selseleh Jonban, Luis Romeral, Mousa Marzband, and Abdullah Abusorrah. A reinforcement learning approach using markov decision processes for battery energy storage control within a smart contract framework. *Journal of Energy Storage*, 86:111342, 2024.

Ping He is an Engineer at State Grid Suzhou Power Supply Company of China. His research interests include flow control and optimization, electric power information network Quality of Service, artificial intelligence, and machine learning.

Dongsheng Jing is a Senior Engineer at State Grid Suzhou Power Supply Company of China. He is majored in computer application technology. His research interests include computer network, electric power information network Quality of Service, intelligent operation and maintenance, artificial intelligence, and machine learning.

Baozhen Qi is a Senior Engineer at State Grid Suzhou Power Supply Company of China. He is majored in computer application technology. His research interests include substation network, flow control and optimization, artificial intelligence, and machine learning.

Yu Yang is a Senior Engineer at State Grid Suzhou Power Supply Company of China. He is majored in software engineering. His research interests include intelligent operation and maintenance, network security, information security, and artificial intelligence.

Jingsong Xue a Senior Engineer at State Grid Suzhou Power Supply Company of China. His research interests include intelligent operation and maintenance, information and communication, electric power information network Quality of Service, network security, and artificial intelligence.

Received: October 15, 2025; Accepted: November 30, 2025.