Multi-Agent Reinforcement Learning-Based Delay and Power Optimization for UAV-WMN Substation Inspection

Qingwei Tang, Wei Sun, *Senior Member, IEEE*, Zhi Liu, *Senior Member, IEEE*, Yang Xiao, *Fellow, IEEE*, Qiyue Li, *Senior Member, IEEE*, Xiaohui Yuan, *Senior Member, IEEE*, and Qian Zhang

***Abstract*—Unmanned aerial vehicles (UAV), due to their flexibility and extensive coverage, have gradually become essential for substation inspections. Wireless mesh networks (WMN) provide a scalable and resilient network environment for UAVs, where each node can serve as either an access point or a relay point, thereby enhancing the network's fault tolerance and overall resilience. However, the UAV-WMN combined system is complex and dynamic, facing the challenge of dynamically adjusting node transmission power to minimize end-to-end (E2E) delay while ensuring channel utilization efficiency. Real-time topology changes, high-dimensional state spaces, and large solution spaces make it difficult for traditional algorithms to guarantee convergence and stability. Generic reinforcement learning (RL) methods also struggle with stable convergence. This paper introduces a new Lyapunov function-based proof to address these issues and provide a stable condition for dynamic control strategies. Then, we developed a specialized neural network power controller and combined it with the MATD3 algorithm, effectively enhancing the system's convergence and E2E performance. Simulation experiments validate the effectiveness of this method and demonstrate its superior performance in complex scenarios compared to other algorithms.**

***Index Terms*—Multi-agent reinforcement learning, wireless mesh networks, neural network, Lyapunov function, RNN, substation inspection.**

Received 30 July 2024; revised 10 February 2025 and 12 March 2025; accepted 2 April 2025. Date of publication 8 April 2025; date of current version 7 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grants 52277087, 62173120, and 52077049; in part by the Natural Science Foundation of Anhui Province under Grants 2108085UD07, 2108085UD11, and 2008085UD04; and in part by the Fundamental Research Funds for the Central Universities under Grant JZ2023HGQA0107. The associate editor coordinating the review of this article and approving it for publication was M. F. Zhani. *(Corresponding author: Wei Sun.)*

Qingwei Tang, Qiyue Li, and Qian Zhang are with the School of Electrical and Automation Engineering, Hefei University of Technology, Hefei 230009, Anhui, China (e-mail: tangqingwei@mail.hfut.edu.cn; liqiyue@mail.ustc.edu.cn; zhangqian@hfut.edu.cn).

Wei Sun is with the School of Electrical and Automation Engineering and the Anhui Engineering Technology Research Center of Industrial Automation, Hefei University of Technology, Hefei 230009, Anhui, China (e-mail: wsun@hfut.edu.cn).

Zhi Liu is with the Department of Computer and Network Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan (e-mail: liu@ieee.org).

Yang Xiao is with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487 USA (e-mail: yangxiao@ieee.org).

Xiaohui Yuan is with the Department of Science and Engineering, University of North Texas, Denton, TX 76207 USA (e-mail: xiaohui.yuan@unt.edu).

Digital Object Identifier 10.1109/TNSM.2025.3558823

I. Introduction

WIRELESS mesh networks (WMN) are a flexible and highly scalable network architecture where each node can not only send and receive data but also serve as a relay to forward data. WMN has shown significant potential in various fields, such as smart cities, smart agriculture, and smart grids [1], [2], [3], [4]. In the field of smart grids, many power companies have begun using UAVs for substation inspections due to their ability to provide high-precision data collection and real-time monitoring [5]. Using WMN-based communication for UAVs can significantly enhance network coverage and data transmission efficiency while increasing system flexibility and reliability. This paper explores the scenario of combining UAVs with WMNs for substation inspections.

In a substation inspection system, UAVs can perform precise inspection tasks for power equipment in complex environments and cover a wide geographic area. On the other hand, the WMN can provide stable data transmission coverage in these complex and changing environments, ensuring reliable communication for UAVs to transmit inspection data. When a node is far from the terminal receiving node, data can be transmitted to the terminal node using a multi-hop method. However, if the transmission power is too low, more hops are required, increasing the end-to-end (E2E) delay. In contrast, increasing the transmission power of a node reduces the number of hops, and thus decreases the E2E delay. However, unrestricted increases in transmission power may cause signal overlap and channel access contention between neighboring nodes, reducing the network's channel efficiency. Additionally, since UAVs act as signal transmission nodes in WMN and their positions constantly change, nodes in WMN must dynamically adjust the network topology accordingly.

The dynamic nature of the UAV-WMN combined system results in a high-dimensional state space and a vast solution space, significantly increasing optimization's complexity and difficulty. Moreover, the UAV's endurance can be negatively impacted by the indefinitely increasing transmission power [6]. Additionally, the UAV-WMN combined system comprises numerous nodes. When one node's state changes, other nodes' states are affected accordingly. This dynamic characteristic makes the system highly sensitive, with an extremely high-dimensional state space, and the control strategy easily influences the system's stability. Therefore, multiple factors must be considered when conducting substation inspections,

including UAV endurance, data return delays, and channel efficiency. The sensitivity of the dynamic system and the stability challenges associated with the high-dimensional state space also need to be addressed. Optimizing such a complex dynamic system is a daunting task, which we refer to as the delay and power optimization (DPO) problem.

A. Related Works

Dynamically adjusting the transmit power of nodes in a WMN system leads to various network topologies. Therefore, the DPO problem can be viewed as a network topology optimization problem. Currently, methods for enhancing network performance by adjusting network topology can be classified into three categories: mathematical planning-based methods, heuristic-based methods, and machine learning-based methods.

1) Mathematical Planning Methods: Mathematical planning based methods offer a rigorous theoretical foundation and precise solutions for WMN topology optimization. For example, integer linear programming has been used to optimize the location and number of nodes for discrete deployment [7]. Mixed integer nonlinear programming can solve complex combinatorial optimization problems, such as optimizing node positions, link quality, and energy efficiency simultaneously [14]. However, while mathematical planning methods are accurate and stable, their computational cost can be prohibitive for large-scale problems, making them unsuitable for mainstream large-scale networks.

2) Heuristic Methods: Heuristic algorithms are widely used for WMN topology optimization due to their simplicity and effectiveness. For instance, the memetic algorithm for topology optimization against cascading failures enhances network robustness and demonstrates excellent time efficiency and topology resilience [9]. A ring topology optimization algorithm has successfully reduced power consumption and improved stability by leveraging a ring configuration [10]. Genetic algorithms optimize network structures for specific applications to minimize delay [11], while ant colony optimization constructs small-world models to reduce delay and enhance stability [12]. The Minimum Spanning Tree heuristic effectively addresses the strong minimum energy topology problem by ensuring connectivity and achieving strong performance [13]. However, heuristic algorithms may converge to local optima, and their performance often relies on parameter tuning through iterative trials.

3) Machine Learning Methods: The popularity of machine learning methods in various fields has prompted researchers to apply them to wireless network topology optimization problems. For example, in the literature [14], the authors propose a two-stage topology-aware deep learning framework. This framework trains a graph embedding unit and a link prediction module and uses them to identify links that might be selected in the optimal network scheduling. Reinforcement learning (RL), a machine learning method where agents learn to make optimal decisions through interactions with their environment, has seen widespread application in robotics and autonomous driving [15], [16]. RL-based optimization algorithms usually do not require accurate modeling of the environment. Therefore, they are highly adaptable and flexible, which is becoming increasingly popular in topology optimization. For instance, in [17], the authors propose a novel deep reinforcement learning (DRL) algorithm for graph search used in network topology optimization. Similarly, in [18], the authors present an efficient DRL-based topology generation algorithm that reduces computational overhead and improves operational efficiency. In [19], the authors introduce a self-organizing network fault management algorithm, where an RL agent adjusts the transmission power of indoor base stations to ensure the signal-to-interference-plus-noise ratio of user equipment meets the target value. Despite RL algorithms not needing precise system modeling, their decision-making and parameter optimization processes involve a degree of randomness, which leads to an unstable convergence process and poor explainability.

Recent studies have explored UAV and RL applications in various scenarios. Reference [41] proposes a UAV-based aerial P2P backbone network using DQN for path planning and energy management, improving data consistency and lookup success in dynamic vehicular ad hoc networks. Reference [42] introduces a DRL-based energy-efficient relay election mechanism, optimizing relay selection via DQN to extend network lifetime and enhance data transmission. Reference [43] presents the MARLF framework, leveraging DRL to optimize UAV trajectories for post-disaster rescue, maximizing coverage, throughput, and energy efficiency while ensuring connectivity. Reference [45] examined machine learning in mobile communications, emphasizing its role in optimization and QoS prediction. It reviews six key studies, showcasing machine learning's potential to enhance system intelligence and efficiency. Notably, [46] introduced the KMV-Cast algorithm, which improves data transmission quality and outperforms other evaluation algorithms.

The primary challenges in using UAV and WMN for substation inspections include real-time changes in environmental topology, high-dimensional state spaces, and extensive solution spaces. These challenges lead to increased E2E delay, higher overall power consumption, and instability in the communication system. Although generalized RL algorithms can address these issues to some extent, several challenges remain. First, the RL training process may be unstable in dynamic and complex environments, resulting in optimization results outside the acceptable range. Second, the inherent uncertainty and low explainability of RL algorithms limit the controllability of decision-making. This paper aims to address the power-delay balance in multi-node WMN systems and ensure the stability of the optimization process.

B. Contributes

For the DPO problem, we propose a stability optimization method based on the multi-agent twin delayed deep deterministic policy gradient (MATD3) algorithm. Our method employs neural networks as power controllers at each node of the WMN to minimize the system's total E2E delay by dynamically adjusting the transmission power of each node. The main contributions are as follows.

(1) We constructed a new discrete Lyapunov function and derived the structural limitations of the stabilizing controller from Theorem 1. This function provides the stability condition for designing the controller.

(2) To achieve stability control, we designed a controller based on the recurrent neural network (RNN) that strictly satisfies the stability conditions. This controller can efficiently handle high-dimensional environmental states, respond effectively to real-time topology changes, and manage sensitive distributed systems.

(3) To validate the effectiveness and practicality of our method, we created several simulation environments modeling realistic scenarios and applied various mainstream MARL methods to them. These experiments confirmed our method's stability and performance advantages in complex real-world environments.

C. Paper Organization

The rest of the paper is organized as follows. Section II details the modeling of the UAV inspection task and the communication delay system, followed by a description of the Markov decision process for DPO and the multi-agent TD3 framework. Section III first introduces the stabilization conditions of the DPO problem, and then describes the control algorithm we designed. In Section IV the results of our numerical study are shown and analyzed. Finally, Section V summarizes the paper.

II. Preliminaries

A. Problem Description

Substation inspection tasks are conducted using UAV to monitor equipment, offering a rapid and safe solution in high-risk, complex environments. This method enhances the reliability and security of substations. Real-time data transmission is essential for timely responses and decision-making during UAV inspections. WMNs are particularly well-suited for this purpose. By interconnecting nodes through wireless links, WMNs create a multi-hop network with advantages such as high reliability, self-organization, and dynamic adaptability. These features enable efficient real-time data backhaul during UAV-based inspections, significantly improving inspection accuracy and efficiency. Fig. 1 illustrates the inspection scenario, with the yellow dashed line representing the UAV's flight trajectory. In realistic substation environments, the presence of numerous power devices and complex surroundings can impose significant communication constraints. To address this, we propose deploying WMN nodes within substations to ensure stable data backhauling. The detailed communication process is depicted in Fig. 2.

This paper simulates real-world environmental conditions by deploying WMN nodes at various (x, y, z) coordinates within a virtual 3D space to support data transmission for a moving UAV. We set up an active node for simulating the UAV in the simulation environment. This UAV acts as a mobile node and moves in a WMN that contains nine fixed nodes and one terminal node. The UAV is responsible for collecting data frames and transmitting these data frames to the fixed

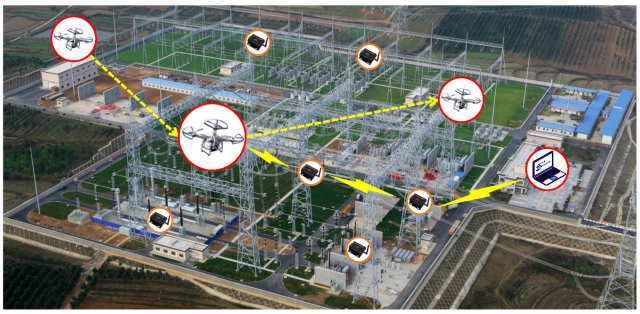


Fig. 1. The UAV-WMN combined inspection system in substations.

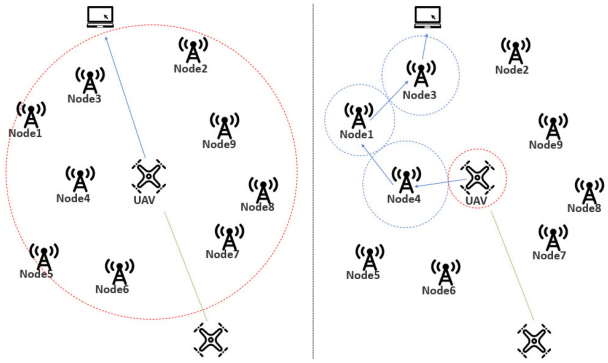


Fig. 2. Two scenarios for transmitting frames during UAV inspections.

nodes in the WMN. Subsequently, these fixed nodes relay the data frames to the terminal node by multi-hop transmission. Through this mechanism, the coverage of the network is effectively extended. We hypothesize:

- All WMN nodes operate on the same channel with a constant signal wavelength.
- The transmission power of each node in the WMN system is adjustable.
- The nodes in the WMN can adjust their transmission power thereby realizing multiple hops and forming different topologies.
- To ensure the consistent performance of each node in the network in terms of reception, the antenna gain and the minimum reception sensitivity $(P_{(min(receive))})$ of all nodes are set to the same constants.
- It is assumed that the transmission quality of all signals in the simulation environment remains stable and the transmission process masking has no effect.
- According to Friis equation, the received power of a node in WMN is: $P_{\text{receive}} = \frac{P_{\text{transmit}} \times G_{\text{transmit}} \times G_{\text{receive}} \times \lambda^2}{(4\pi d)^2} \geq P_{\min(\text{receive})}$, where P_{receive} is the received power, P_{transmit} is the transmit power, G_{transmit} is the transmit antenna gain, G_{receive} is the receive antenna gain, λ is the signal wavelength, d is the distance between the transmitter node and the receiver node, and $P_{\min(\text{receive})}$ is the receive sensitivity. To achieve the desired minimum receive sensitivity, the communication distance can be extended by adjusting the transmit power and optimizing the network topology.

Based on these assumptions, we observe a direct positive relationship between the transmitting power of nodes and

their communication range. By regulating the transmit power of WMN and UAV nodes, it is possible to control their communication ranges, construct diverse network topologies, and significantly influence overall network performance. Given the limited battery life of UAVs, maintaining a low yet sufficient transmit power to sustain communication can effectively reduce energy consumption and extend airborne operation time. This study examines the scenario in which a UAV transmits data frames to a terminal node while in motion. During the UAV's flight, all nodes in the WMN, including the UAV, dynamically adjust their transmit power in real time to optimize communication delays. The goal is to achieve an optimal balance between transmit power and system delay. In this context, the cooperative operation of all nodes and maintaining a dynamic trade-off between delay and power present a challenging optimization problem.

We illustrate the problem in Fig. 2. The green dashed line depicts the UAV's motion trajectory, while the red dashed circle represents its communication range. The blue arrows indicate the data transmission path from the UAV to the end node. In the scenario on the left of Fig. 2, the UAV directly communicates with the end node using maximum transmit power. Although this approach is straightforward, it significantly reduces the UAV's battery life, compromising its ability to perform prolonged monitoring and patrolling tasks. Conversely, in the scenario on the right of Fig. 2, the UAV delivers data packets to the end node through a multi-hop transmission path involving nodes 1, 4, and 3. By leveraging pre-deployed WMN nodes, the UAV can dynamically regulate its transmit power to enable efficient multi-hop data delivery. Simultaneously, each WMN node adjusts its transmit power in real-time to maintain stable communication with the UAV. This mechanism ensures real-time data transmission to the end node during the UAV's flight, optimizes communication delay, and enhances energy efficiency and overall communication performance.

In short, the transmission power level selected by the UAV when sending data frames to end nodes significantly impacts the overall network performance. Low transmission power may require data frames to be relayed through intermediate nodes, increasing the number of transmission hops and end-to-end (E2E) delay. Conversely, high transmission power reduces delay by enabling direct communication between the UAV and the end nodes. However, high power usage consumes more channel resources and can disrupt other devices in the WMN, such as those communicating between nodes 7 and 8 or nodes 9 and 10, which must wait for the channel to become available. The interaction between the WMN topology and network performance is highly complex and cannot be captured by a simple function.

B. Communications System Modeling

1) Transmission Delay Modeling: The CSMA/CA algorithm in the IEEE 802.15.4 protocol facilitates wireless communication by minimizing data collisions. It achieves this by first verifying that the channel is idle before transmission and introducing a random backoff time to enhance

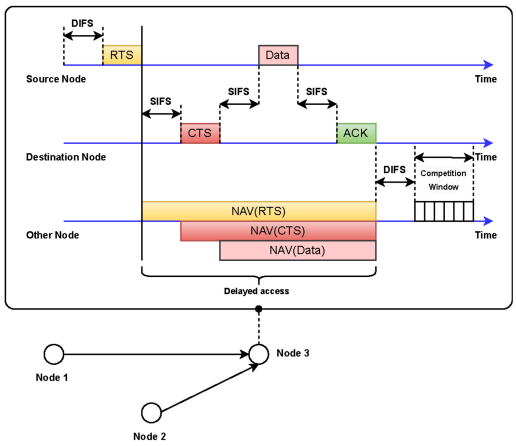


Fig. 3. WMN environment model.

the efficiency and reliability of data transmission [22]. Since CSMA/CA operates as a stochastic process, a Markov chain model is suitable for simulating the data transmission behavior of nodes in a wireless network. In this study, we assume a uniform data arrival rate λ for all nodes, with all nodes sharing the same channel. Fig. 3 illustrates the delay calculation process for communication between nodes 1 and 2 with node 3. Based on the backoff mechanism described in [23], and as expressed in Eq. (1) and Eq. (2), we construct a corresponding Markov model.

$$P_g = P\{t < T_{\text{slot}}\} = 1 - e^{-\lambda T_{\text{slow}}} \tag{1}$$

$$\alpha = 1 - \left(1 - P_g\right)^{Ng} \tag{2}$$

where P_g represents the likelihood of a node in the idle state producing at least one packet within a single time slot. The parameter λ indicates the packet arrival rate, measured as the number of packets arriving per second. T_{slot} denotes the duration of a time slot, while α signifies the probability that the channel is currently occupied. Additionally, N_g refers to the count of neighboring nodes within the transmission range of the current node.

The channel contention access delay at the MAC layer can be derived from the steady state probability distribution equations of the Markov chain model, which was constructed in the literature [23] for the IEEE802.15.4 protocol CSMA/CA, as shown in Eq. (3).

$$\begin{cases} T_{bi} = \sum_{j=0}^{W_i-1} j\omega_{(i,j)} t_s, i \in [0, H] \\ T_{mac} = \sum_{i=0}^{H} \left(T_{bi} \sum_{j=0}^{2^{i+3}-1} \omega_{(i,j)} \right) \end{cases} \tag{3}$$

where T_{bi} represents the average time spent by the node in the ith fallback phase; t_s is the time of state transfer, which is a fixed value; and T_{mac} denotes the channel contention access delay at the MAC layer. The total WMN delay is shown in Eqs. (4), (5), (6).

$$T_{tx} = \frac{\mathrm{L}}{v} \tag{4}$$

$$P_f = P_g(1-\alpha)^{Ng} \tag{5}$$

$$T_{\text{delay}} = T_{\text{mac}} + \left(1 - P_f\right) T_{tx} \tag{6}$$

where L represents the size of the packet, v is the bandwidth of the wireless communication, P_f denotes the probability

of transmission failure, and T_{tx} is the time consumed for transmitting the packet along the channel. Therefore, the transmission time of node T_{delay} can be derived.

The congestion between WMN communication links is correlated, so their queues can be modeled by using the *M*/*M*/1 queuing model proposed in queuing theory [21]. This model assumes that the nodes follow the Poisson process, where the packet arrival rate is λ and the packet service rate is $\mu = 1/T_{mac}$. According to the *M*/*M*/1 queueing model, the queuing delay of a node can be expressed as $T_q = \frac{\lambda}{\mu(\mu-\lambda)}$. The single-hop delay mainly consists of transmission time and queuing delay, i.e., $T_{node} = T_{delay} + T_q$. Therefore, we can further derive the E2E delay from the source node to the aggregation node, $T_{ete} = \sum T_{node}$.

2) Radiant Power Modeling: The UAV inspection process considers the sustainability of each device within the WMN. The WMN communication system must overcome challenges posed by the natural environment and address background noise generated by power electronics and communication equipment. Additionally, the masking effect of the signal propagation path may lead to multi-path propagation of electromagnetic waves, involving phenomena such as direction changes, reflection, scattering, and diffraction. To accurately analyze the transmission characteristics of wireless signals, researchers frequently employ the logarithmic path loss model. This model effectively represents the relationship between electromagnetic wave signals and transmission paths in wireless networks while supporting sustainable development objectives. According to the model, the received signal strength at a node is approximated by the logarithmic path loss formula shown in Eq. (7).

$$P_r = P_t - P_L(d_0) - 10\log_{10}\left(\frac{d}{d_0}\right) - X_\sigma \tag{7}$$

where P_r represents the received signal strength at the receiving node, while P_t denotes the transmission power of the sending node. The term $P_L(d_0)$ represents the reference path loss at a predefined distance d_0 (typically 1 meter). The actual separation between the sender and receiver is given by d, measured in meters. The path loss exponent s characterizes the attenuation rate with increasing distance. Additionally, X_σ accounts for multipath effects, modeled as a Gaussian random variable with mean 0 and variance σ_X^2, i.e., $X_\sigma \sim N(0, \sigma_X^2)$. In this study, we assume a constant received signal strength P_r at each node. The relationship between transmission power and distance follows the logarithmic path loss model, as expressed in Eq. (8).

$$P_t = P_r + 10 \times s \times \log_{10}\left(\frac{d}{d_0}\right) + P_L(d_0) + X_\sigma \tag{8}$$

C. Communication Delay and Transmit Power Correlation Modeling

To evaluate the communication performance under different topologies, we use Eq. (9) to express the relationship between the transmit power and the system delay according to the literature [25], [26].

$$\mathcal{T}(\mathcal{P}) = \sum_{i=0}^{3} (\mathcal{Q}_f)^i (1-\mathcal{Q}_f)\left[i\mathcal{T}_f(\boldsymbol{p}_k^i) + \mathcal{T}_s(\boldsymbol{p}_k^i)\right] + (\mathcal{Q}_f)^4 4\mathcal{T}_f(\boldsymbol{p}_k^i) \tag{9}$$

where $\boldsymbol{p}_t^i$ represents the transmit power value of node i at moment k, $\mathcal{Q}_f = (1-\zeta)^{Ng}$ represents the probability of frame transmission failure, and ζ represents the probability of Node sending a frame. $\mathcal{T}_s(\boldsymbol{p}_k^i)$ represents the delay generated by a successful state transfer from node i to the next node, and $\mathcal{T}_f(\boldsymbol{p}_k^i)$ represents the delay generated by a failed state transfer from node i. The values of $\mathcal{T}_s(\boldsymbol{p}_k^i)$ and $\mathcal{T}_f(\boldsymbol{p}_k^i)$ can be calculated by using Eqs. (10), (11).

$$\mathcal{T}_s\left(\boldsymbol{p}_k^i\right) = \mathcal{T}_b\left(\boldsymbol{p}_k^i\right) + SIFS + ACK + DIFS + \delta \tag{10}$$

$$\mathcal{T}_f\left(\boldsymbol{p}_k^i\right) = \mathcal{T}_b\left(\boldsymbol{p}_k^i\right) + DIFS + \delta + \mathcal{T}_m\left(\boldsymbol{p}_k^i\right) \tag{11}$$

where $\mathcal{T}_b(\boldsymbol{p}_k^i)$ is the frame transmission delay from node i to the next node when using transmit power $\boldsymbol{p}_k^i$. $\mathcal{T}_m(\boldsymbol{p}_k^i)$ is the average backoff delay from node i to the next node when using transmit power $\boldsymbol{p}_k^i$. δ represents the propagation delay of the signal from the sender to the receiver. The rest of the parameters are expressed in Fig. 3, and the detailed derivations are given in the literature [25], [26]. Bringing Eqs.(10) (11) into Eq. (9) yields Eq. (12).

$$\begin{aligned}
\mathcal{T}(\mathcal{P}) &= \sum_{i=0}^{3} (Q_f)^i (1-Q_f)\left[i\mathcal{T}_f\left(p_k^i\right) + \mathcal{T}_s\left(p_k^i\right)\right] + (Q_f)^4 4\mathcal{T}_f\left(p_k^i\right) \\
&= \sum_{i=0}^{3} (Q_f)^i (1-Q_f)\left[i\mathcal{T}_b\left(p_k^i\right) + i(\mathrm{DIFS}+\delta) + i\mathcal{T}_m\left(p_k^i\right)\right. \\
&\quad \left. + \mathcal{T}_b\left(p_k^i\right) + \mathrm{SIFS} + \mathrm{ACK} + \mathrm{DIFS} + \delta\right] \\
&\quad + (\mathcal{Q}_f)^4 4\left(\mathcal{T}_b\left(\boldsymbol{p}_k^i\right) + \mathrm{DIFS} + \delta + \mathcal{T}_m\left(\boldsymbol{p}_k^i\right)\right) \\
&= \sum_{i=0}^{3} (Q_f)^i (1-Q_f)\left[(i+1)\mathcal{T}_b\left(\boldsymbol{p}_k^i\right) + (i+1)(\mathrm{DIFS}+\delta)\right. \\
&\quad \left. + i\mathcal{T}_m\left(\boldsymbol{p}_k^i\right) + \mathrm{SIFS} + \mathrm{ACK}\right] \\
&\quad + (\mathcal{Q}_f)^4 4\mathcal{T}_b\left(\boldsymbol{p}_k^i\right) + (\mathcal{Q}_f)^4 4(\mathrm{DIFS}+\delta) + (\mathcal{Q}_f)^4 4T_m\left(\boldsymbol{p}_k^i\right)
\end{aligned} \tag{12}$$

Further combining yields Eq. (13).

$$\begin{aligned}
\mathcal{T}(\mathcal{P}) &= \left(\sum_{i=0}^{3} (Q_f)^i (1-Q_f)(i+1) + (Q_f)^4 4\right)\mathcal{T}_b\left(p_k^i\right) \\
&\quad + \left(\sum_{i=0}^{3} (Q_f)^i (1-Q_f) i + (Q_f)^4 4\right)\mathcal{T}_m\left(p_k^i\right) \\
&\quad + \sum_{i=0}^{3} (Q_f)^i (1-Q_f)[(i+1)(\mathrm{DIFS}+\delta) + \mathrm{SIFS} + \mathrm{ACK}] \\
&\quad + (Q_f)^4 4(\mathrm{DIFS}+\delta)
\end{aligned} \tag{13}$$

Then we approximate Eq. (13) as $\mathcal{T} = (X+Y)\mathcal{P} + Z + V = H\mathcal{P} + \mathcal{T}^{env}$ in this paper our dynamic equation for delay control can be expressed as shown in Eq. (14) and (15).

$$\mathcal{T}(k+1) = \boldsymbol{H}\mathcal{P}(k+1) + \mathcal{T}^{env} \tag{14}$$

$$\mathcal{P}(k+1) = \mathcal{P}(k) + \Delta T \cdot u_i\left(p_k^i, T\right), \forall i \in \mathcal{N} \quad (15)$$

To ensure that the designed control algorithm is stable, it is crucial to find a controller $\mathbf{u} = -g_\theta(\boldsymbol{p}_k^i, \mathcal{T})$ and a Lyapunov function V that satisfies the stability condition in Proposition 1. Since the controller inputs depend on the states $\boldsymbol{p}_k^i$ and $\mathcal{T}$, the system dynamics can be approximated as $\mathcal{T}$(k+1) = $\mathcal{T}(k) - \boldsymbol{H}\, g_\theta(\boldsymbol{p}_k^i, \mathcal{T})$: = $f_u(\mathcal{T}(k))$. We designed the Lyapunov function of Eq. (16) to help us prove stability, where A is the unit matrix satisfying positive definiteness.

$$V(\mathcal{T}) = \left(\mathcal{T} - f_u(\mathcal{T})\right)^\top A^{-1}\left(\mathcal{T} - f_u(\mathcal{T})\right) \quad (16)$$

In this paper's WMN environment, the node set is $\{n_1, n_2, \ldots, n_N\}$, with corresponding E2E delay $\{T_{\text{ete}}^1, T_{\text{ete}}^2, \ldots, T_{\text{ete}}^N\}$ and transmit power $\{P_t^1, P_t^2, \ldots, P_t^N\}$. We define T as the maximum E2E delay across all nodes and E as the average transmit power. Since E2E delay and transmit power have different units, they are normalized according to Eqs. (17) and (18). All experimental specifications are presented in normalized form. The purpose of this approach is to facilitate the observation of various indicators during the training process.

$$\text{NormT}_1 = \frac{T - T_{min}}{T_{max} - T_{min}} \quad (17)$$

$$\text{NormP}_2 = \frac{E - E_{min}}{E_{max} - E_{min}} \quad (18)$$

D. Markov Decision Process for DPO

This paper aims to require all nodes in a WMN system to adjust their transmit power to minimize the E2E delay. We formulate the problem as Dec-POMDP, denoted as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, \text{R}, \gamma)$. The state space, observation space, action space, and reward function are defined as follows.

(1) State space: $\mathcal{S} = \mathcal{D} \times \mathcal{P}$, where $\mathcal{D}$ and $\mathcal{P}$ represent the set of communication delays of the system and the transmit power of each node, respectively. In the $\mathcal{T}$-set, the total system delay of the WMN can be denoted as $\mathcal{D} = \{\sum_{i=1}^n t_k^i : t_k^i \in (0, d)\}$. In the set $\mathcal{P}$, the transmit power of each node in the WMN can be expressed as $\mathcal{P} = \{(p_k^1, \ldots, p_k^i, \ldots, p_k^n) : p_k^i \in (-p, p)\}$, where p_t^i denotes the value of the transmit power of the node i at the k moment.

(2) Observation space: the observation space of each agent is constructed based on the transmit power and the total system delay of each agent in the previous moment in the WMN, which is a subset of the global observation. In this paper, we define $o_i \in \mathcal{O}$ as $p_k^i, \mathcal{D}$, where p_k^i is the transmit power of the ith agent; D is the total system delay. Observation space is used for actual input and decision-making by agents.

(3) Action space: in this paper, we deploy neural networks on each WMN node and UAV node, considering them as agents in the RL task. According to the dynamic equation of delay control $\mathcal{P}(k+1) = \mathcal{P}(k) + \Delta T \cdot u_i(p_k^i, T)$, for each agent i, we use the neural network to generate a power increment based on the previous moment $\mathcal{A} : \mathcal{A}_i = \{\Delta a : 0 \le \Delta a \le c\}$, where c represents the maximum transmitting power that can be generated. It is denoted as $a_i^{k+1} = a_i^k + \Delta a$.

(4) Reward Function: To balance the E2E delay by adjusting the node transmission power in the WMN, for agent i, we take the delay $\mathcal{T}$ as the main factor of reward, and the node transmission power p_k^i as a secondary factor. Therefore, the reward function can be defined in Eq. (19).

$$\mathcal{R} = \alpha e^{-\mathcal{D}} - \beta \sum_{i=1}^n ln\left(\boldsymbol{p}_k^i\right) - \omega \sum_{i=1}^n \mathbb{I}\left(\mathbf{p}_k^i > p_{\max}\right) \quad (19)$$

where α is the weight of the delay, representing the degree that we expect to minimize the delay. $\mathcal{D}$ represents the total system delay and is computed as: $\mathcal{D} = \sum_{i=1}^n t_k^i$, ω is a regularization term for excessive power. $\mathbb{I}(\boldsymbol{p}_k^i > p_{\max})$ is a barrier function that returns 0.99 when the transmit power of node i exceeds the maximum permissible value of p_{max}, and otherwise returns 0. The weighting factors α, β, ω are set empirically, we set α to 0.6, β to 0.3, and ω to 0.1.

(5) Objective Function: In summary, in this paper, we aim to minimize the total system delay by adjusting the transmit power of each node under the constraints, as shown in Eq. (20). We are interested in optimizing the communication performance of the system during UAV inspections with a dynamic, real-time method.

$$\begin{cases} max_{\theta_i} \sum_{i=1}^n \mathcal{R} \\ min_{\Delta a_i^k} \sum_k \sum_{i=1}^n \boldsymbol{t}_k^i \\ s.t. \mathcal{T}(k+1) = H\mathcal{P}(k+1) + \mathcal{T}^{env} \\ \mathcal{P}(k+1) = \mathcal{P}(k) + \Delta T \cdot u_i\left(\boldsymbol{p}_k^i, \mathcal{T}\right) \\ u_i(t) = -g_{\theta_i}(v_i(t)) \\ \boldsymbol{p}_k^i \in (-p, p) \end{cases} \quad (20)$$

E. Multi-Agent TD3 Framework

Value-based DRL algorithms, such as proximal policy optimization (PPO) [28], soft actor-critic (SAC) [29], asynchronous advantage actor-critic (A3C) [30], and deep deterministic policy gradient (DDPG) [31], excel in sample efficiency and stability for real-world tasks with continuous action spaces. These algorithms consist of an actor network for policy generation and a critic network for evaluating policy performance. MATD3 [44], an improved actor-critic (AC) algorithm, addresses non-stationary multi-agent problems in continuous action spaces. It features decentralized (online and target) actor networks, a centralized (online and target) critic network, an empirical replay buffer, and exploration noise, ensuring stable and efficient learning. The centralized critic network learns the value function $Q^\pi(\mathcal{S}, \mathcal{A})$ by minimizing Eq. (21).

$$L\left(\theta^Q\right) = \mathbb{E}_{(s,a,s',r)\sim D}[(Q_{\theta^Q}(s, a_1, \ldots, a_n) - \left(r + \gamma Q_{\theta^{Q'}}\left(s', \pi_i\left(s_1'\right), \ldots, \pi_i\left(s_n'\right)\right)\right))^2] \quad (21)$$

where θ^Q and $\theta^{Q'}$ are the parameters of the critic network and target critic network, respectively, π_i is the strategy of agent i, γ is the discount factor, and D is the experience pool. And the Decentralized actor network is used to make actions based on local observations. Its parameters θ^{π_i} are updated by maximizing Eq. (22).

$$\nabla_{\theta^{\pi_i}} J(\theta^{\pi_i}) \approx \mathbb{E}_{s\sim D, \epsilon\sim\mathcal{N}}\Big[\nabla_a Q_{\theta^Q}(s, a_1, \ldots, a_n)\Big|_{a_i=\pi_{\theta^{\pi_i}}(s_i)+\epsilon} \quad \nabla_{\theta^{\pi_i}} \pi_{\theta^{\pi_i}}(s_i)\Big] \quad (22)$$

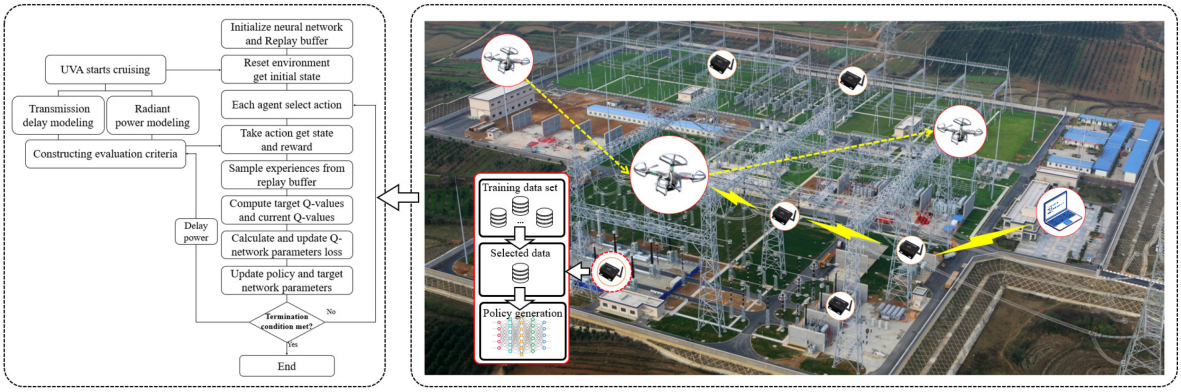


Fig. 4. Overview of DPO problem and solution methods.

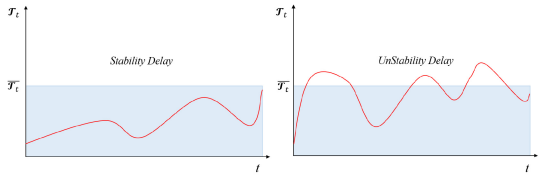


Fig. 5. Delay stability of WMN.

where $J(\theta^{\pi_i})$ is the strategy gradient of agent i and ϵ is the strategy noise for increasing exploration. To increase exploration and stability, MATD3 introduces the strategy noise $a_i = \pi_{\theta^{\pi_i}}(s_i) + \epsilon$ in the actor network update. in this paper, the parameters of the target network are updated as shown in Eq. (23).

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\kappa)\theta^{Q'}; \theta^{\pi_i'} \leftarrow \tau\theta^{\pi_i} + (1-\kappa)\theta^{\pi_i'} \quad (23)$$

III. Method

In this chapter, we will systematically show specific methods for solving the DPO problem. Fig. 4 illustrates the problem addressed in this paper and the method flowchart. In Section III-A, we will discuss the stability condition for the DPO problem. In Section III-B, we will present a method for constructing a neural network that satisfies the stability condition. In Section III-C, we will describe the learning and optimization process of the algorithm in detail.

A. DPO System Stabilization Conditions

To demonstrate the stability of the DRL algorithm employed to address the DPO problem, the exploration space of the strategy is confined to the set of stable controllers, guided by Lyapunov stability theory, as depicted in Fig. 5. Specifically, this paper utilizes LaSalle's invariance principle, an extension of Lyapunov's method, to derive the algorithm's stability conditions. First, we define the concept of DPO delay stability.

Definition 1 (Time-Delay Stability): A closed-loop system is considered stable if, for any time delays $\mathcal{T}^{env}$ and initial condition $\mathcal{T}(0)$, the trajectory $\mathcal{T}(t)$ eventually converges to the set $S_{\mathcal{T}} = \{\mathcal{T} \in \mathbb{R}^n : 0 \le \mathcal{T}(t) \le \overline{\mathcal{T}t}\}$. This means that $\lim_{t\to\infty} \mathrm{dist}(\mathcal{T}(t), S_{\mathcal{T}}) = 0$, where the distance function is defined as $\mathrm{dist}(\mathcal{T}(t), S_{\mathcal{T}}) = \min_{\mathcal{T}' \in S_{\mathcal{T}}} |\mathcal{T}(t) - \mathcal{T}'|$.

Proposition 1 (LaSalle's Theorem for Discrete-Time Systems [32]): For the discrete-time system $x(t+1) = f(x(t))$, suppose there exists a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying the following conditions:

1) $V(x) \ge 0$ for all $x \in \mathbb{R}^n$.
2) Along the trajectories of the system, the Lyapunov function V is non-increasing, i.e., $V(f(x)) - V(x) \le 0$.

Define the set E be the set of points where $V(f(x)) = V(x)$, i.e., $E = \{x \in \mathbb{R}^n \mid V(f(x)) - V(x) = 0\}$. Define M as the largest invariant set within E. If there exists a constant $a \in \mathbb{R}^+$ such that the level set $L_a = \{x \mid V(x) \le a\}$ is bounded, then for any initial state $x(0) \in L_a$, the trajectory $x(t)$ will eventually approach the set M. Furthermore, if V is radially unbounded, the trajectory $x(t)$ will also converge to M for any initial state $x(0) \in \mathbb{R}^n$.

The stability of the DPO problem can be ensured by choosing a controller $\mathbf{u} = -g_\theta(p_k^i, \mathcal{T})$ and a Lyapunov function V that satisfies the stability condition in Proposition 1. For the delay control system described by Eq. (20), since the control input $\mathbf{u} = -g_\theta(p_k^i, \mathcal{T})$ depends on the state $\mathcal{T}$, the closed-loop system dynamics are given by $\mathcal{T}(k+1) = \mathcal{T}(k) - Hg_\theta(p_k^i, \mathcal{T}) := f_u(\mathcal{T}(k))$. We adopt the Lyapunov function specified in Eq. (24).

$$V(\mathcal{T}) = (\mathcal{T} - f_u(\mathcal{T}))^{\mathcal{T}} A^{-1} (\mathcal{T} - f_u(\mathcal{T})) \quad (24)$$

According to LaSalle's theorem in Proposition 1, for the function V, if $V(f_u(\mathcal{T})) - V(\mathcal{T}) \le 0$, and if and only if $\mathcal{T}$ belongs to the delay stability set $S_{\mathcal{T}}$, $V(f_u(\mathcal{T})) - V(\mathcal{T}) = 0$. Here, $S_{\mathcal{T}}$ is defined as the delay stability set, that is $S_{\mathcal{T}} = \{\mathcal{T} \in \mathbb{R}^n \mid \underline{\mathcal{T}_t} \le \mathcal{T}_t \le \overline{\mathcal{T}_t}\}$. For any initial delay configuration $\mathcal{T}(0) \in \mathbb{R}^n$, the trajectory $\mathcal{T}(t)$ will eventually converge to the largest invariant set within $S_{\mathcal{T}}$. Additionally, if for each agent i, the control action satisfies $u_i = 0$ when $\mathcal{T}_t$ lies within the interval $[\underline{\mathcal{T}_t}, \overline{\mathcal{T}t}]$, then $S_{\mathcal{T}}$ itself becomes an invariant set.

We give a sufficiently necessary condition for the above property to hold in Theorem 1, which ensures the stability of the delay. The objective of this paper is to design the controller $\mathbf{u} = -g_\theta(p_k^i, \mathcal{T})$, which is approximated in writing as $\mathbf{u} = -g_\theta(\mathcal{T})$, such that Lyapunov function satisfies the following two properties.

1) $V(f_u(\mathcal{T})) - V(\mathcal{T}) < 0, \quad \forall \mathcal{T} \notin S_{\mathcal{T}}$
2) $V(f_u(\mathcal{T})) - V(\mathcal{T}) = 0 \quad$ for $\quad \mathcal{T} \in S_{\mathcal{T}}$

Theorem 1 (Delay Stability Condition): Suppose that for all nodes i, the function $g_{\theta_i}(\cdot)$ is continuously differentiable and satisfies $u_i = -g_{\theta_i}(\mathcal{T}) = 0$ when $\mathcal{T}_t \in [\underline{\mathcal{T}_t}, \overline{\mathcal{T}_t}]$. In the interval $\mathcal{T}_t \in [\overline{\mathcal{T}_t}, \infty)$, each $\frac{\partial g_{\theta_i}}{\partial \mathcal{T}}$ satisfies Eq. (25). And $\lim_{|\mathcal{T}|\to\infty} |g_{\theta_i}(\mathcal{T})| = \infty$ when $|\overline{\mathcal{T}_t}| \to \infty$. Then, the delay stability in Definition 1 holds.

$$-\frac{2}{\Delta \mathcal{T}} A^{-1} \prec \frac{\partial u}{\partial \mathcal{T}} \prec 0 \tag{25}$$

Theorem 1 Derivation: According to the Lyapunov function of Eq. (24), to ensure the stability of the system delay, we need to prove that the Lyapunov function $V(\mathcal{T})$ is non-increasing as in Eq. (26).

$$\frac{dV(\mathcal{T})}{d\mathcal{T}} = 2(\mathcal{T} - f_u(\mathcal{T}))^T A^{-1} \left(\frac{\partial \mathcal{T}}{\partial \mathcal{T}} - \frac{\partial f_u(\mathcal{T})}{\partial \mathcal{T}} \right) < 0 \tag{26}$$

where

1) $\frac{\partial f_u(\mathcal{T})}{\partial \mathcal{T}} : \frac{\partial f_u(\mathcal{T})}{\partial \mathcal{T}} = I - H \frac{\partial g_\theta(\mathcal{T})}{\partial \mathcal{T}}$
2) $\frac{\partial g_\theta(\mathcal{T})}{\partial \mathcal{T}} : \frac{\partial u}{\partial \mathcal{T}} = -\frac{\partial g_\theta(\mathcal{T})}{\partial \mathcal{T}}$

Subsequently, we substitute $\frac{\partial f_u(\mathcal{T})}{\partial \mathcal{T}}$ into the expression for the derivative of the Lyapunov function, and then obtain Eq. (27).

$$\begin{aligned} \frac{dV(\mathcal{T})}{d\mathcal{T}} &= 2(\mathcal{T} - f_u(\mathcal{T}))^T A^{-1} \left(I - \left(I - H \frac{\partial g_\theta(p_k, \mathcal{T})}{\partial \mathcal{T}} \right) \right) \\ &= 2(\mathcal{T} - f_u(\mathcal{T}))^T A^{-1} H \frac{\partial g_\theta(p_k, \mathcal{T})}{\partial \mathcal{T}} \end{aligned} \tag{27}$$

To ensure the stability of system delay, it is necessary to satisfy Eq. (28).

$$2A^{-1} H \frac{\partial g_\theta(p_k, \mathcal{T})}{\partial \mathcal{T}} < 0 \to -\frac{2}{\Delta T} A^{-1} < 0 \tag{28}$$

According to the literature [33] and Eq. (25), the stability condition can be expressed as $\frac{\partial \mathbf{u}}{\partial \mathcal{T}} < 0$ when the sampling time $\Delta T \to 0$. The left side of Eq. (25) is naturally satisfied, we just need to focus on the right-hand side condition. Due to the decentralized nature of MARL, $\frac{\partial \mathbf{u}}{\partial \mathcal{T}}$ is a diagonal matrix as show in Eq. (29). To satisfy the delay stability, we design each $\frac{\partial g_{\theta_i}}{\partial \mathcal{T}} > 0$ to be strictly monotonically increasing.

$$\frac{\partial \mathbf{u}}{\partial \mathcal{T}} = -\begin{bmatrix} \frac{\partial g_{\theta_1}}{\partial \mathcal{T}_1} & \cdots & 0 \\ \ddots & & \ddots \\ 0 & \cdots & \frac{\partial g_{\theta_n}}{\partial \mathcal{T}_n} \end{bmatrix} \tag{29}$$

B. Design of Neural Network Controllers for DPO

To satisfy the stability requirements outlined in Theorem 1, we developed a novel algorithm based on MATD3. The controller must adhere to a strictly monotonic decreasing condition according to the delay stability criterion. Therefore, we employed monotonic functions to design the policy function in the RL framework. Previous studies, such as literature [36], have proposed methods for constructing specialized monotonic neural networks.

In this paper, we introduce two dedicated neural networks, $\xi^+(x; w^+, b^+)$ and $\xi^-(x; w^-, b^-)$, both constructed using single-layer networks with ReLU activation functions. For $\xi^+(x; w^+, b^+)$, let the weight vector for agent i be denoted as $q_i = [q_i^1, q_i^2, \ldots, q_i^m]$, with corresponding bias vector $b_i = [b_i^1, b_i^2, \ldots, b_i^m]^\top$. For $\xi^-(x; w^-, b^-)$, let $z_i = [z_i^1, z_i^2, \ldots, z_i^m]$ represent the weight vector, and $c_i = [c_i^1, c_i^2, \ldots, c_i^m]^\top$ the bias vector. Additionally, let $l \in \mathbb{R}^m$ denote a column vector of ones. A detailed construction of both networks is provided in Lemma 1.

Lemma 1: Let $\mu(x) = \max(x, 0)$ denote the ReLU function. The stacked ReLU function defined by Eq. (30) remains zero for $x \le 0$ and increases monotonically for $x > 0$. Similarly, the version defined by Eq. (31) is zero when $x \ge 0$ and exhibits a monotonically increasing behavior for $x < 0$.

$$\begin{cases} \xi^+(x; w^+, b^+) = q_i \mu(1x + b^+) \\ \sum_{l=1}^{d'} w_l^+ > 0, \forall d' = 1, \ldots, d, b_1^+ = 0, b_l^+ \le b_{l-1}^+, \forall l = 2, \ldots, d \end{cases} \tag{30}$$

$$\begin{cases} \xi^-(x; w^-, b^-) = z_i \mu(-1x + b^-) \\ \sum_{l=1}^{d'} w_l^- > 0, \forall d' = 1, \ldots, d, b_1^- = 0, b_l^- \le b_{l-1}^-, \forall l = 2, \ldots, d \end{cases} \tag{31}$$

It is worth noting that while Theorem 1 ensures the necessary condition for asymptotic stability of the system, it does not directly guarantee stability within a finite time. To achieve exponential stability, it is necessary to strengthen the Lyapunov condition, specifically in the form $V(\mathcal{T}_{t+1}) - V(\mathcal{T}_t) \le -cV(\mathcal{T}_t)$, where $0 < c < 1$. In this case, the stability condition can be expressed as: $-\frac{1+\sqrt{1-c}}{\Delta T} A^{-1} < \frac{\partial \mathbf{u}}{\partial \mathcal{T}} < -\frac{1-\sqrt{1-c}}{\Delta T} A^{-1}$. Typically, we can find a relatively small ΔT to ensure that the trained policy satisfies the above inequalities. The controller proposed in this paper achieves stability of the state.

The RNN is a class of neural networks specifically designed for processing sequential data. Their capacity to handle temporal sequences makes them particularly suitable for policy networks in RL tasks. RNNs effectively capture the temporal relationships and dynamic state transitions, enabling improved decision-making. To satisfy the stability requirements outlined in Theorem 1, we propose a novel RNN architecture incorporating a monotonic neural network structure and fully connected layers. This architecture employs a monotonically increasing activation function to ensure system stability and adherence to the reinforcement Lyapunov condition. In accordance with the constraints specified in Eq. (25), the policy controller in our RL framework is designed to be monotonically increasing. Specifically, the hidden state update function of the RNN is formulated based on the neural networks defined in Eq. (30) and Eq. (31), as illustrated in Eq. (32).

$$h_t = \xi_{\theta^i}^+(\mathcal{T}_t - \overline{\mathcal{T}_t}) + \xi_{\theta^i}^-(\mathcal{T}_t - \underline{\mathcal{T}_t}) \tag{32}$$

where $\xi_{\theta^i}^+(\mathcal{T}_t) : \mathbb{R} \to \mathbb{R}$ is monotonically increasing when $\mathcal{T}_t > \overline{\mathcal{T}_t}$ and equal to zero when $\mathcal{T}_t \le \overline{\mathcal{T}_t}$. Similarly, $\xi_{\theta^i}^-(\mathcal{T}_t - \underline{\mathcal{T}_t}) : \mathbb{R} \to \mathbb{R}$ is monotonically increasing when $\mathcal{T}_t < \underline{\mathcal{T}_t}$, and zero otherwise. We approximate parameterize the controller at node i as $g_{\theta_i}(\mathcal{T}) = [\xi_{\theta^i}^+(\mathcal{T}_t - \overline{\mathcal{T}_t}) + \xi_{\theta^i}^-(\mathcal{T}_t - \underline{\mathcal{T}_t})]$. Because $\mathbf{u} = -g_\theta(\mathcal{T})$, $\frac{\partial \mathbf{u}}{\partial \mathcal{T}} \prec 0$ is satisfied.

C. Learning and Optimization

We deploy neural networks on each WMN and UAV node, treating them as agents in the RL task. Specifically, first we use the online policy network to generate the current action $a_{j+1} = \mu(s_{j+1};\theta_{\text{now}}) + \epsilon$ where ϵ satisfies the truncated normal distribution, and $\epsilon \sim \mathcal{N}(0,\sigma^2,-c,c)$ is used to introduce exploration. In this paper, we set $\sigma = 0.05$.

Then we use two online critic networks, each with a corresponding target network, for a total of four networks. The smaller value from the assessments is used to mitigate overestimation. We denote the parameters of the two online networks as ϕ_1 and ϕ_2, respectively, and the parameters of their target networks as ϕ_1^- and ϕ_2^-. The objective function of the critic network is shown in Eq. (33).

$$\begin{cases} \mathcal{L}(\phi_1) = \mathbb{E}_{(s,a,r,s',d)\sim D}\Big[\Big(Q_{\phi_1}(s,\mu_\theta(s)) - \\ \quad \Big(r + \gamma \min\limits_{i=1,2} Q_{\phi_i^-}\big(s',\mu_\theta^-(s')\big)\Big)\Big)^2\Big] \\ \mathcal{L}(\phi_2) = \mathbb{E}_{(s,a,r,s',d)\sim D}\Big[\Big(Q_{\phi_2}(s,\mu_\theta(s)) - \\ \quad \Big(r + \gamma \min\limits_{i=1,2} Q_{\phi_i^-}\big(s',\mu_\theta^-(s')\big)\Big)\Big)^2\Big] \end{cases} \tag{33}$$

To effectively control the variation amplitude of the transmit power of each node in the WMN and enhance the stability of the training process, we introduce the method of KL scatter regularization to constrain the strategy network, and the specific objective function of the strategy network is shown in Eq. (34).

$$\begin{aligned} \mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s',d)\sim D}\Big[&- \min Q_{\phi_i}(s,\mu(s;\theta)) \\ &+\alpha\, KL[\pi_{\text{old}}(\cdot|s)\,||\,\pi_{\text{new}}(\cdot|s)]\Big] \end{aligned} \tag{34}$$

where α is the coefficient that controls the strength of regularization of KL scattering, we set it to 0.1. The policy network is updated in the following way. The network parameters are updated as shown in Eq. (35). The detailed flow of the algorithm is shown in Algorithm 1.

$$\begin{cases} \phi_1^- = \rho\phi_1^- + (1-\rho)\phi_1 \\ \phi_2^- = \rho\phi_2^- + (1-\rho)\phi_2 \\ \mu_\theta^- \leftarrow \tau\mu_\theta + (1-\tau)\mu_\theta^- \end{cases} \tag{35}$$

Algorithm 1 MARL Algorithm for DPO Problem

Initialize policy/target policy network parameters θ, θ', Q-network/target Q-network parameters ϕ_1,ϕ_2, ϕ_1', ϕ_2', and replay buffer $\mathcal{D}$
for agent $i = 1, 2, \ldots N$ **do**
  Initialize actor network θ with random parameters
  Initialize target networks $\theta' \leftarrow \theta$
end for
Set global time step $T \leftarrow 0$
for *episode* $= 1$ to T **do**
  for time step $t = 1, 2, \ldots, N$ **do**
    Interact with environment and obtain a transition (s, a, r, s', d) in $\mathcal{D}$
    Sample mini-batch of N transitions (s, a, r, s', d) from $\mathcal{D}$
  end for
  Update global time step $T \leftarrow T + 1$
  for agent $i = 1, 2, \ldots, N$ **do**
    Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon, \epsilon \sim N(0, \sigma)$
    Observe Reward r and new state s' store transition tuple (s, a, r, s') in $\mathcal{D}$
    for $j = 1 \rightarrow N_c$ **do**
      For each $\tau \sim (s_i, a_i, r, s_i', d)$ in batch calculate the target Q: $y_i = r + \gamma \min_{i=1,2} Q_{\phi_i}^-\left(s', \mu_\theta^-(s')\right)$
      Calculate the online Q: $Q_{\phi_1}(s, \mu_\theta(s))$ and $Q_{\phi_2}(s, \mu_\theta(s))$
      Update the parameters ϕ_1, ϕ_2 of critic using (33) generated transitions
    end for
    for $j = 1 \rightarrow N_a$ **do**
      Preventing overestimation using smaller Q values $minQ_{\phi_i}(s, \mu(s;\theta))$
      Compute value of KL regularization
      Update the parameters θ of actor using (34) generated transitions
    end for
    Soft-update the parameters of target critic network and target actor network using (35)
  end for
end for

IV. Experiments

A. Experimental Settings

We generated a node distribution map with dimensions of $50m \times 50m \times 10m$, where the positions of each node are shown in Fig. 6. There are a total of 11 nodes, with node 0 being the terminal node. The algorithm in this paper is based on the MATD3 algorithm, using the Adam optimizer to update neural network parameters. The simulated experiments were performed on a workstation equipped with an NVIDIA® GeForce RTX 4080 16GB GPU and an Intel® Core™ i7-13700KF processor (clocked at 3.40 GHz). All MARL algorithms were implemented using PyTorch, with hidden neurons set to 64.

Additionally, we set communication environment parameters based on the recommendations in literature [34], and some defined environment parameters are listed in Tab. I. For detailed parameter settings, refer to [34]. The neural network used in this paper is lean and efficient, containing 2560 parameters and performing 8648 floating point operations. It can achieve fast and efficient operations in environments with limited computational resources. Before conducting the experiments, we placed two fixed nodes at the positions furthest and closest to the terminal node in a straight line and measured the total delay of the WMN system at these positions. The delays are 0.15 and 0.4, respectively. We set the delay upper limit slightly below the average value. The technical indicators (power and delay) in this paper have been normalized. Detailed normalization methods and parameter units can be found in [35].

B. Four Cases for DPO

To validate the effectiveness of the proposed algorithms, we reproduce several mainstream MARL algorithms and apply them to the DPO environment, including Multi-agent deep deterministic policy gradient (MADDPG), Counterfactual multi-agent policy gradients (COMA), Multi-agent proximal

TABLE I
EXPERIMENTAL PARAMETERS

Parameter name	Value	Parameter name	Value
ζ	0.01	SIFS	$28\mu s$
δ	$1.5\mu s$	DIFS	$128\mu s$
ρ	0.05	ACK	$50\mu s$
τ	0.05	Backoff exponent	4
Actor Learning Rate	0.001	Frame size	256 bits
Critic Learning Rate	0.01	Random Seed	100, 200, 300
Max Step	200	$\overline{T_t}, \overline{T_t}$ for case1,2,3	(0, 0.265)
Max Episode	1000	$\overline{T_t}, \overline{T_t}$ for case4	(0, 0.270)
Parameters	256	FLOPs	8648

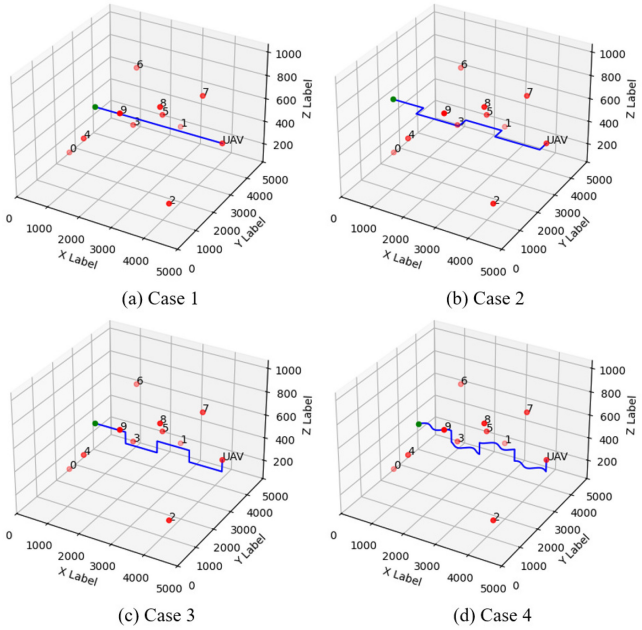


Fig. 6. Examples of four types of UAV inspections.

policy optimization (MAPPO), and Multi-agent twin delayed deep deterministic policy gradient (MATD3). It is worth noting that our algorithms are improved based on the MATD3 algorithm. Specifically, MADDPG is an extension of DDPG for multi-agent environments, handling collaborative and competitive tasks through centralized training and decentralized execution; COMA reduces the variance in policy gradients by using a centralized critic and counterfactual baselines, improving optimization efficiency and stability; MAPPO is a multi-agent version of PPO, supporting shared or independent policy networks and providing training stability through a centralized critic; MATD3 reduces bias and over-estimation in Q-value estimation by introducing delayed updates and dual Q-networks, making it suitable for continuous control tasks in multi-agent systems.

To realistically simulate UAV operations in real-world environments, we deployed 9 wireless communication nodes, 1 UAV node, and 1 terminal node (numbered 0) in 3D space. The coordinates of the UAV node are dynamically adjusted based on training steps. Fig. 6 illustrates three common UAV flight modes in distribution network inspections: Case 1 is the straight-line cruise mode, suitable for open, obstacle-free areas, efficiently covering large ground areas. Case 2 is the flat-fluctuation cruise mode, ideal for detailed

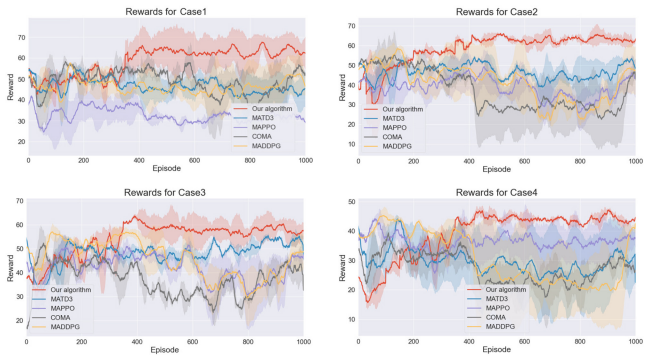


Fig. 7. Training Reward for Four Cases.

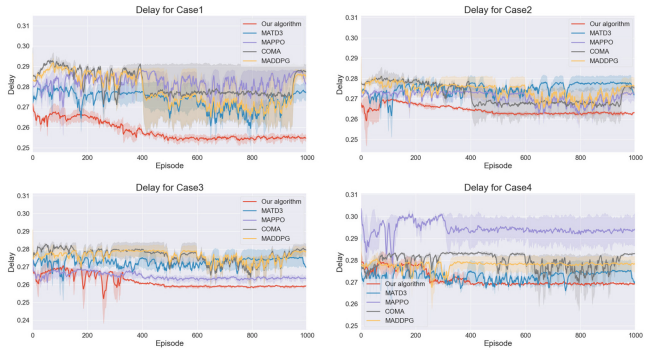


Fig. 8. Training Delay for Four Cases.

inspections in specific areas. Case 3 is the vertical cruise mode, designed for monitoring tasks at different altitudes. Additionally, Case 4 simulates specific tasks by adding random fluctuations along the Z-axis, reflecting real-world requirements such as obstacle avoidance. This setup enhances the realism of the environment simulation.

Fig. 7 details the reward changes during the training of the DPO problem under four cruising modes. In Case 1, our algorithm demonstrates increasing stability and higher average rewards as training progresses, confirming its incremental improvement and long-term stability. In Case 2, it achieves higher rewards with smaller fluctuations in later stages, proving its effectiveness and reliability in complex environments. Case 3 maintains high rewards, showcasing strong adaptability and robustness in vertical cruising. In Case 4, despite a general decrease in maximum rewards across all algorithms, our algorithm performs well, maintaining high rewards in most stages, further validating its robustness and effectiveness in variable environments.

Fig. 8 illustrates the convergence of training delays (in seconds) for different algorithms under four cruising modes. In Case 1, our algorithm converges the fastest with stable trends, achieving a convergence delay approximately 5.6% lower than comparative algorithms, which show greater fluctuations. In Case 2, it maintains minimal delay, highlighting its advantage in planar cruising, with a convergence delay approximately 3.8% lower than others. In Case 3, our algorithm achieves the lowest delay and fastest convergence, with a delay approximately 3% lower than comparative algorithms. In Case 4, despite delay reductions across all algorithms, ours maintains

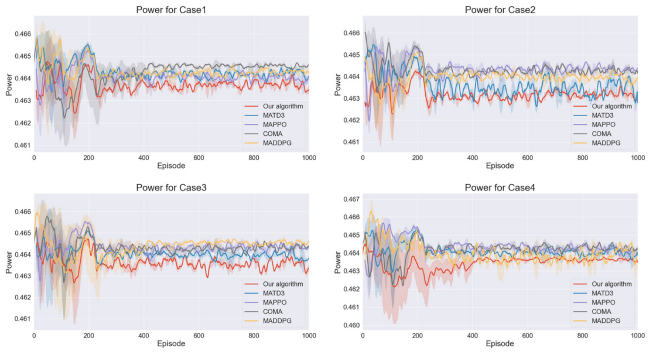


Fig. 9. Training Transmission Power for Four Cases.

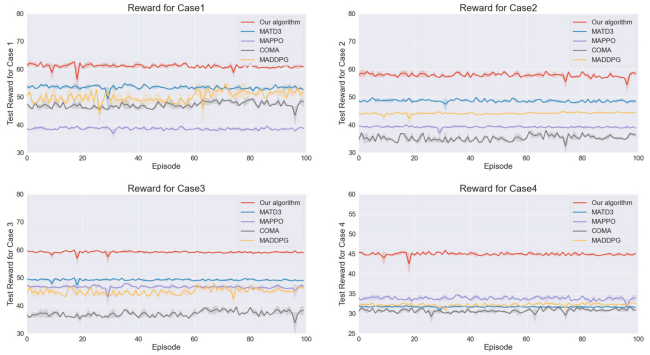


Fig. 10. Testing Reward for Four Cases.

the lowest delay with minimal fluctuations, approximately 2.8% lower than others. These results demonstrate that our algorithm consistently achieves lower delays and better stability across all cruising modes, showcasing its effectiveness and robustness in the DPO problem.

To evaluate power optimization (in dBm), we compared our algorithm with others in Fig. 9. Across Cases 1 to 4, our algorithm consistently achieves superior power optimization, minimizing power output while maintaining low delay. These results highlight its excellence in power management, suggesting potential benefits for extending device battery life and improving system response speed in practical applications.

To evaluate the effectiveness of our algorithms, we conducted comprehensive tests on all algorithms in a unified environment, focusing on Reward, communication delay, and power consumption. To test adaptability and stability in rapidly changing environments, we increased the UAV node update rate by 10 times while keeping other parameters unchanged. Fig. 10 shows that our algorithm consistently achieves significantly higher rewards than comparative algorithms across all scenarios. In contrast, some comparative algorithms exhibited large reward fluctuations, indicating weaker robustness. Fig. 11 presents test results for power and delay. Our algorithm achieves optimal delay and power across cases, with delays 4.6%, 3.2%, 1.2%, and 4.7% lower than the best comparative algorithms in Cases 1 through 4, respectively. Additionally, it reduces transmission power while maintaining low communication delay, demonstrating excellent energy efficiency and delay optimization, making it particularly suitable for resource-constrained devices like UAVs.

C. Effect of Interfering Factors on Performance

Signal interference: Such as environmental noise and radio frequency interference, reduces the packet delivery ratio (PDR). In simulations, we lower the PDR to emulate interference and evaluate algorithm performance in the complex Case 4 scenario. While communication quality is generally high during inspections, interference occurs unpredictably. Introducing interference during training may cause overfitting and hinder the learning of optimal strategies under normal conditions. Thus, we introduce interference only in the testing phase to assess robustness. When the PDR drops below 70%, communication quality degrades, causing unreliable data transmission. We test PDR levels of 90%, 80%, and 70%, while the PDR in the original setup was set to 99%. As shown in Fig. 12, interference begins at the 20th episode during testing, affecting system delay. Results show that greater interference increases delays and fluctuations. Nevertheless, the algorithm restores stability effectively. At 90% PDR, delays remain minimal with smooth curves, indicating strong stability. At 80% PDR, delays increase slightly with minor fluctuations, quickly stabilizing. At 70% PDR, delays rise significantly with pronounced fluctuations, but the system still recovers, demonstrating robustness.

Node Failure: To evaluate the impact of node failures on the performance of the proposed algorithm, we designed various node failure scenarios for simulation testing. Specifically, we randomly remove a relay node during the training and testing phases to simulate real-world node failure scenarios. This approach allows us to verify the system's self-healing capability. Unlike interference, node failure has a clearly defined triggering mechanism, making it easier to simulate in a planned manner. In Fig. 13, each subfigure presents the training results for Cases 1 through 4. When node failures occurred (highlighted by closed boxes), there were significant fluctuations in delay. However, the model quickly adapted to the new topology using our method. The delay gradually recovered and stabilized, demonstrating the algorithm's learning ability and adaptability in dynamic topologies. In Fig. 14, node failures were also introduced during the testing phase for Cases 1 through 4. The results show that, although the failures caused a temporary delay increase, the model stabilized the network within a few episodes. The delay rapidly decreased and remained stable, showcasing strong fault-tolerance capabilities.

D. Effect of Node Count on Performance

Several studies have validated strategies to enhance scalability and stability in complex networks. For example, [39] proposed an E2E communication scheme for large-scale urban transportation networks, while [40] used a time-triggered model to improve wireless communication reliability in environments like large hospitals. Our method has been validated with 11 nodes, including one UAV and ten fixed nodes. We trained the algorithm in networks with larger node counts to assess scalability and robustness. By gradually increasing nodes under the most complex Case 4 and observing Reward, Delay, and Power during convergence, we compared

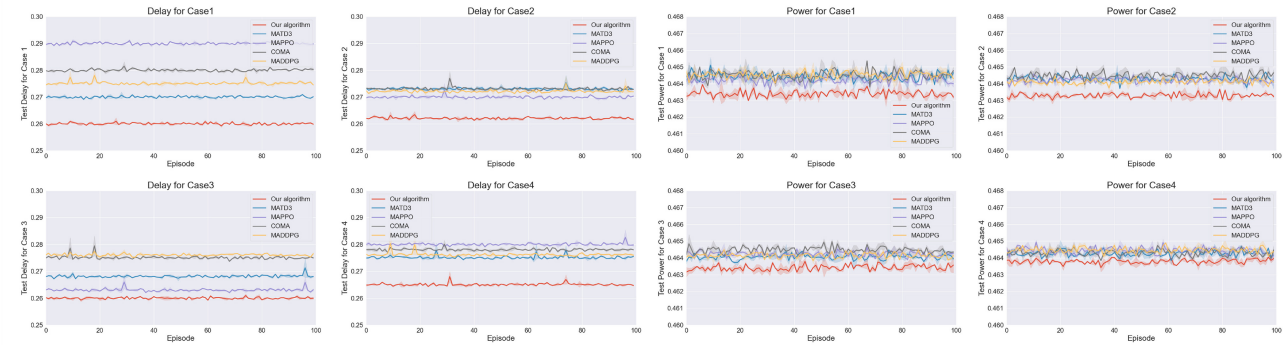


Fig. 11. Testing Delay and Transmission Power for Four Cases.

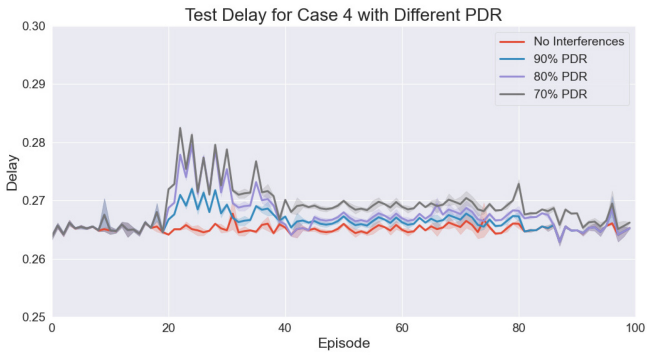


Fig. 12. The Effect of Interferences on Case 4 Testing Delay.

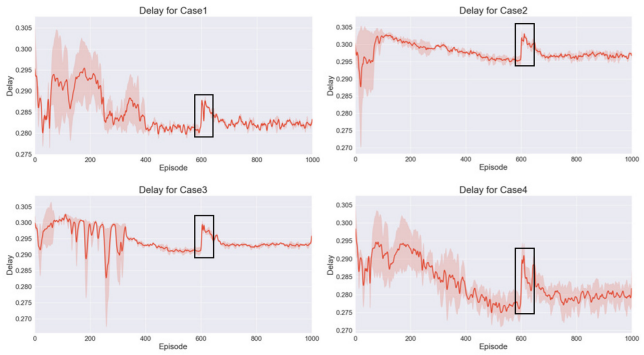


Fig. 13. Node Failure Training Delay for Four Cases.

our algorithm with other mainstream MARL methods. The results show that, within the same space, our algorithm consistently achieves the highest Reward and optimal power-delay performance as node count increases. While a moderate increase in nodes optimizes communication paths, reducing delay and improving rewards, excessive nodes raise topological complexity, increasing delay and reducing rewards. Fig. 15 illustrates a nonlinear relationship between node count and rewards, where adding nodes does not always improve system performance. Furthermore, excessive nodes cause communication redundancy, increase power consumption, and conflict with green energy goals.

E. Centralized and Distributed Performance Evaluation

DQN is a classical centralized algorithm that combines deep learning with reinforcement learning, utilizing a deep neural

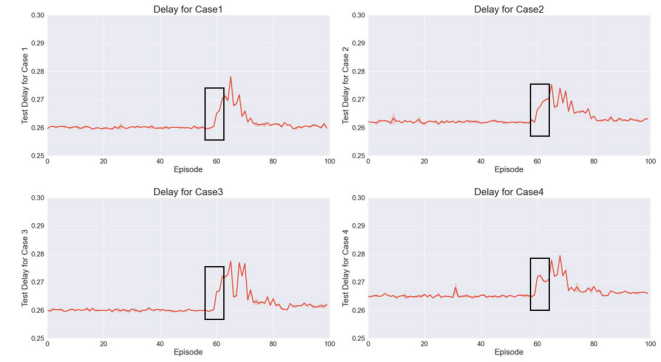


Fig. 14. Node Failure Testing Delay for Four Cases.

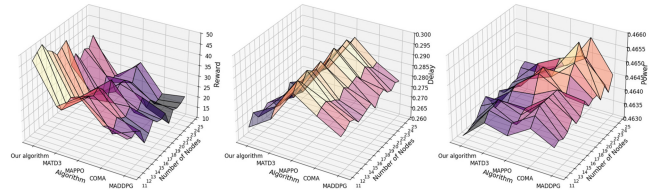


Fig. 15. Convergence Training Reward, Delay, and Power for Different Numbers of Nodes.

network to approximate the Q-value function and guide the agent in selecting optimal actions. To evaluate the performance of DQN and our algorithm, we present the reward values for both methods across Cases 1 to 4. As illustrated in Fig. 16, our algorithm consistently outperforms DQN in all scenarios. This is primarily attributed to the distributed algorithm's ability to harness multi-agent collaboration and parallelism, mitigating the single-point failure risks inherent in centralized methods through autonomous decision-making. Furthermore, during the testing phase, our algorithm not only delivers superior performance but also demonstrates greater stability compared to DQN. These experimental results validate the distributed algorithm's superior adaptability and robustness in substation inspection tasks.

F. Effect of Different Node Placements on Performance

In practical applications, the deployment locations of relay nodes are often influenced by site conditions and network requirements, which may lead to dynamic changes. We set

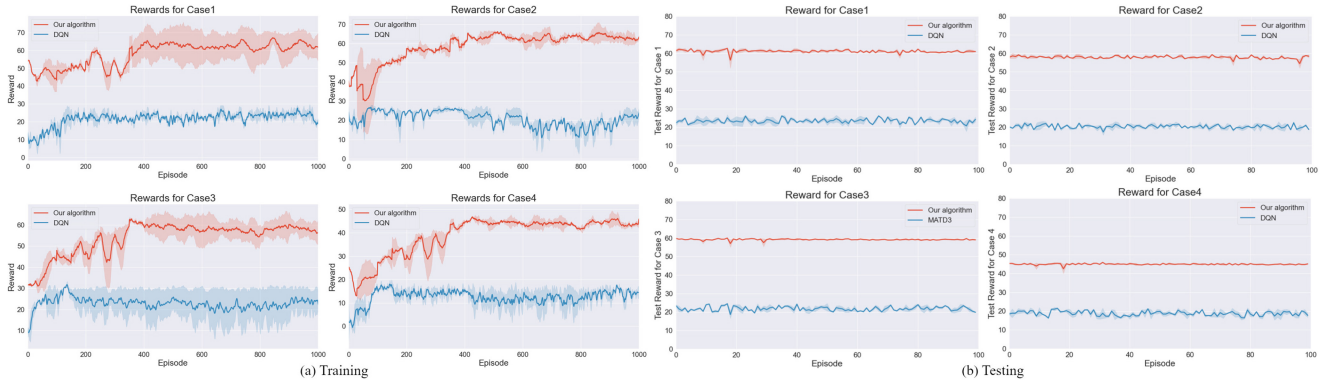


Fig. 16. Training/Testing Reward of Centralized and Distributed Algorithms for Four Cases.

TABLE II
COORDINATES OF RELAY NODES IN DIFFERENT CONFIGURATIONS

Set	Configuration Coordinates
A	[2523, 4664, 204], [4472, 482, 306], [1928, 2995, 408], [1460, 1167, 510], [2974, 2784, 620], [1341, 4199, 735], [4022, 3106, 810], [3646, 1447, 920], [3243, 92, 900]
B	[2033, 529, 667], [2249, 2831, 607], [1720, 4385, 663], [4095, 4336, 546], [724, 1500, 670], [375, 1222, 68], [3613, 1976, 655], [2547, 4072, 809], [1658, 1411, 375]
C	[3920, 1525, 891], [1856, 3231, 668], [3020, 1867, 711], [1316, 4674, 549], [1661, 4522, 318], [588, 2602, 631], [2703, 2370, 252], [809, 3746, 221], [322, 4626, 552]
D	[2913, 3419, 915], [3329, 1375, 754], [3235, 4120, 111], [4255, 2353, 208], [3453, 2832, 993], [704, 2208, 360], [4432, 2421, 227], [1058, 4442, 281], [2820, 2209, 717]

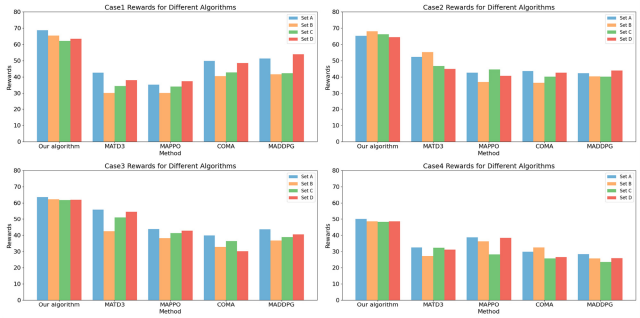


Fig. 17. Convergent Training Rewards of Different Node Configuration Coordinates for Four Cases.

multiple groups of relay nodes with different spatial coordinates to evaluate the performance of each algorithm, using the converged reward value as the evaluation criterion. The coordinates of the relay nodes in different configurations are shown in the Tab. II. Where Set A is the standard node configuration. As shown in Fig. 17, regardless of whether in simple or complex scenarios, the proposed algorithm consistently achieves higher converged rewards with minimal fluctuation compared to the other algorithms. This is attributed to the stability mechanism incorporated into the controller, enabling it to adapt to dynamic environments and tolerate node location changes. In contrast, the reward of other algorithms exhibits significant fluctuations, revealing their limited adaptability and robustness.

G. Performance Comparison and Module Contribution Analysis

To better illustrate the performance differences, Fig. 18 (a) presents the metrics of each algorithm at convergence. Through numerical comparison, our algorithm demonstrates its advantages in communication delay, power consumption, and reward value. In Case 1, our algorithm reduced communication delay by up to 9.02%, achieved a maximum single-step energy saving of 0.22%, and consistently outperformed the comparison algorithms in terms of reward. It effectively balances delay and energy efficiency while performing exceptionally well in complex environments. To demonstrate each module's contribution to system performance, we incrementally add modules under the most complex Case 4, observing the convergence values of key metrics and comparing the system's performance at different stages, and the results are shown in Fig. 18 (b). As shown in Fig. 18 (b), the performance improvements of each system module are as follows: Baseline + A represents the addition of the RNN module, Baseline + B represents the inclusion of stability control based on the RNN, and Baseline + B + C represents the addition of policy constraints on top of the stability control. Using Reward as the analysis criterion, the reward increase for Baseline + A is approximately 9.5%; for Baseline + B, it is approximately 41.0%; and for Baseline + B + C, it is approximately 9.02%. This demonstrates that each module in our algorithm is effective, with the largest performance gain coming from the stability RNN controller design.

V. CONCLUSION

This paper proposes a MARL-based DPO method to address communication challenges in UAV-assisted substation inspections. Our algorithm uses neural networks as distributed agents at each WMN node to dynamically adjust transmission power and minimize E2E delay. It considers the transmission power of both the UAV and WMN nodes and overall network channel utilization. A key innovation is the development of a novel Lyapunov function and an RNN-based power controller, which manages nonlinear environmental changes and ensures system stability. Simulations show that our algorithm outperforms existing algorithms in terms of stability and performance under realistic conditions. By integrating RL into topology optimization, we achieve a dynamic balance between transmission delay and power consumption, enhancing system performance for high-reliability and efficiency scenarios.

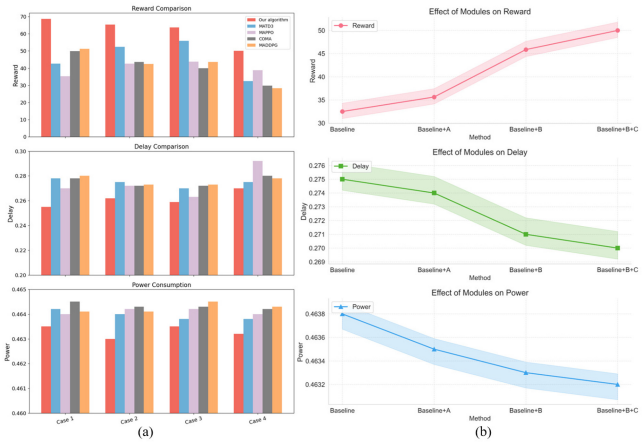


Fig. 18. Comparison of Reward, Delay, and Power Consumption Across Different Algorithms and Module Configurations.

In future work, we aim to extend the scalability of the proposed algorithm to accommodate diverse operational environments and more complex network configurations, further broadening its applicability to real-world scenarios.

APPENDIX A
PROOF OF THEOREM 1

Recall the closed-loop delay dynamics:

$$\mathcal{T}(t+1) = f_u(\mathcal{T}(t)) \tag{36}$$

where the control input is defined as $\mathbf{u}(t) = -g(\mathcal{T}(t))$. Define:

$$h(\mathcal{T}(t)) = \mathcal{T}(t) - f_u(\mathcal{T}(t)) \tag{37}$$

The Lyapunov function is given by:

$$V(\mathcal{T}(t)) = h(\mathcal{T}(t))^T A^{-1} h(\mathcal{T}(t)) \tag{38}$$

Next, we express $h(\mathcal{T}_{t+1})$ in terms of $h(\mathcal{T}_t)$ as follows:

$$h(\mathcal{T}_{t+1}) = h(\mathcal{T}_t) + \int_0^1 \frac{\partial h}{\partial \mathcal{T}}(\mathcal{T}_t + \tau(\mathcal{T}_{t+1} - \mathcal{T}_t))(\mathcal{T}_{t+1} - \mathcal{T}_t)\, d\tau \tag{39}$$

According to Kowalewski's Mean Value Theorem ([37, Th. 1]), we have:

$$h(\mathcal{T}_{t+1}) = h(\mathcal{T}_t) + J_h(\mathcal{T}_{t+1} - \mathcal{T}_t) \tag{40}$$

where:

$$J_h = \sum_{i=1}^{n} \lambda_i \frac{\partial h}{\partial \mathcal{T}}(\mathcal{T}_t + k_i(\mathcal{T}_{t+1} - \mathcal{T}_t)) \tag{41}$$

for $k_i \in [0,1]$, $\lambda_i \geq 0$ for all i, and $\sum_{i=1}^{n} \lambda_i = 1$.

Noting that $\mathcal{T}_{t+1} - \mathcal{T}_t = f_u(\mathcal{T}_t) - \mathcal{T}_t = -h(\mathcal{T}_t)$, we get:

$$\begin{aligned} h(\mathcal{T}_{t+1}) &= h(\mathcal{T}_t) + J_h(\mathcal{T}_{t+1} - \mathcal{T}_t) \\ &= h(\mathcal{T}_t) + J_h(-h(\mathcal{T}_t)) \\ &= h(\mathcal{T}_t) - J_h h(\mathcal{T}_t) \\ &= (I - J_h)h(\mathcal{T}_t) \end{aligned} \tag{42}$$

Therefore, the Lyapunov function at time $t+1$ is:

$$\begin{aligned} V(\mathcal{T}_{t+1}) &= h(\mathcal{T}_{t+1})^T A^{-1} h(\mathcal{T}_{t+1}) \\ &= ((I - J_h)h(\mathcal{T}_t))^T A^{-1}((I - J_h)h(\mathcal{T}_t)) \\ &= h(\mathcal{T}_t)^T (I - J_h)^T A^{-1}(I - J_h)h(\mathcal{T}_t) \end{aligned} \tag{43}$$

We express the Jacobian of the closed-loop delay dynamics as:

$$G(\mathcal{T}, \theta) = \frac{\partial f_u}{\partial \mathcal{T}} + \frac{\partial f_u}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathcal{T}} \tag{44}$$

and define:

$$J_G = \sum_{i=1}^{n} \lambda_i G(\mathcal{T}_t + k_i(\mathcal{T}_{t+1} - \mathcal{T}_t), \theta) \tag{45}$$

where k_i and λ_i follow the definition of J_h. From the definitions of J_h and $h(\mathcal{T}_t)$, we have:

$$J_G = I - J_h \tag{46}$$

Therefore, we obtain:

$$V(\mathcal{T}_{t+1}) - V(\mathcal{T}_t) = h(\mathcal{T}_t)^T \left(J_G^T A^{-1} J_G - A^{-1} \right) h(\mathcal{T}_t) \tag{47}$$

Using Jensen's inequality [38], for $x \in \mathbb{R}^n$, we further have:

$$\begin{aligned} x^T J_G^T A^{-1} J_G x &= \left\| A^{-1/2} J_G x \right\|^2 \\ &= \left\| \sum_{i=1}^{n} \lambda_i A^{-1/2} G(\mathcal{T}_t + k_i(\mathcal{T}_{t+1} - \mathcal{T}_t), \theta) x \right\|^2 \\ &\leq \sum_{i=1}^{n} \lambda_i \left\| A^{-1/2} G(\mathcal{T}_t + k_i(\mathcal{T}_{t+1} - \mathcal{T}_t), \theta) x \right\|^2 \\ &= \sum_{i=1}^{n} \lambda_i x^T G(\mathcal{T}_t + k_i(\mathcal{T}_{t+1} - \mathcal{T}_t), \theta)^T A^{-1} \\ &\quad G(\mathcal{T}_t + k_i(\mathcal{T}_{t+1} - \mathcal{T}_t), \theta) x \end{aligned} \tag{48}$$

Thus, for all $\mathcal{T} \in \mathcal{X}$, if $G(\mathcal{T}, \theta)^T A^{-1} G(\mathcal{T}, \theta) - A^{-1} < 0$, then:

$$V(\mathcal{T}_{t+1}) - V(\mathcal{T}_t) < 0 \tag{49}$$

which means the Lyapunov function is decreasing along the system trajectory.

Finally, recall $g_{t,\theta_1}(\mathcal{T}) = 0$ for $\mathcal{T} \in [\underline{\mathcal{T}}, \overline{\mathcal{T}}]$, so $V(\mathcal{T}_{t+1}) - V(\mathcal{T}_t) = 0$ implies $\mathcal{T}_t \in S_{\mathcal{T}}$.

Given $G(\mathcal{T}, \theta) = I + I_{\Delta T} A \frac{\partial \mathbf{u}}{\partial \mathcal{T}}$, the stability condition becomes:

$$\left(I + I_{\Delta T} A \frac{\partial \mathbf{u}}{\partial \mathcal{T}} \right)^T A^{-1} \left(I + I_{\Delta T} A \frac{\partial \mathbf{u}}{\partial \mathcal{T}} \right) - A^{-1} < 0 \tag{50}$$

Due to the diagonal nature of $\frac{\partial \mathbf{u}}{\partial \mathcal{T}}$, expanding the multiplication terms, we get the stability condition as:

$$-\frac{2}{\Delta T} A^{-1} < \frac{\partial \mathbf{u}}{\partial \mathcal{T}} < 0 \tag{51}$$

By LaSalle's Invariance Principle and the fact that $\lim_{r \to \infty} \|g_\theta(\mathcal{T})\| = \infty$, the stability constraint is summarized in Theorem 1. Proof completed.

APPENDIX B
PROOF OF LEMMA 1

To construct a monotonically increasing function $\xi^+(x; w^+, b^+)$, we can utilize stacked ReLU functions $\mu(x)$, where the ReLU function is defined as $\mu(x) = x$ for $x > 0$ and $\mu(x) = 0$ for $x \leq 0$. Define the functions $\alpha_i^l(\omega_i) = q_i^l \mu(\omega_i + b_i^l)$, where q_i^l are weights and b_i^l are biases, with $b_i^1 = 0$ and $b_i^l \leq b_i^{l-1}$. This implies that as the layer index l increases, the biases do not increase and may decrease.

Due to the decreasing biases b_i^l, each function $\alpha_i^l(\omega_i)$ activates sequentially. Specifically, as ω_i increases, the first function activated is $\alpha_i^1(\omega_i) = q_i^1 \mu(\omega_i)$ because $b_i^1 = 0$. As ω_i further increases beyond $-b_i^2$, $\alpha_i^2(\omega_i)$ starts to activate, i.e., $\alpha_i^2(\omega_i) = q_i^2 \mu(\omega_i + b_i^2)$, producing a non-zero output. This sequential activation continues until $\alpha_i^m(\omega_i)$.

By summing these functions, we obtain a piecewise linear function:

$$\xi^+\left(x; w^+, b^+\right) = \sum_{l=1}^{m} \alpha_i^l(\omega_i) = \sum_{l=1}^{m} q_i^l \mu\left(\omega_i + b_i^l\right) \quad (52)$$

where each segment's slope is $\sum_{j=1}^{l} q_i^j$. To ensure monotonicity, each segment's slope must be non-negative, i.e., $\sum_{j=1}^{l} q_i^j \geq 0$ for all $1 \leq l \leq m$.

Similarly, we can construct a monotonically decreasing function $\xi^-(x; w^-, b^-)$. For negative weights w_i, define $\alpha_i^l(\omega_i) = z_i^l \mu(\omega_i + b_i^l)$, where z_i^l are weights. To maintain monotonicity, each segment's slope must be non-positive, i.e., $\sum_{j=1}^{l} z_i^j \leq 0$ for all $1 \leq l \leq m$.

REFERENCES

[1] L. Mamatas, V. Demiroglou, S. Kalafatidis, S. Skaperas, and V. Tsaoussidis, "Protocol-adaptive strategies for wireless mesh smart city networks," *IEEE Netw.*, vol. 37, no. 2, pp. 136–143, Mar./Apr. 2023.

[2] M. Pan, C. Chen, X. Yin, and Z. Huang, "UAV-aided emergency environmental monitoring in infrastructure-less areas: LoRa mesh networking approach," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2918–2932, Feb. 2022.

[3] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A survey on LoRa for smart agriculture: Current trends and future perspectives," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3664–3679, Feb. 2023.

[4] Q. Tang, W. Sun, Z. Liu, Q. Li, and X. Yuan, "Multi-agent reinforcement learning based dynamic self-coordinated topology optimization for wireless mesh networks," *J. Netw. Comput. Appl.*, vol. 239, Jul. 2025, Art. no. 104177, doi: 10.1016/j.jnca.2025.104177.

[5] Q. Jiang, Y. Liu, Y. Yan, X. Mao, H. Xu, and X. Jiang, "BIM-based 3-D multimodal reconstruction for substation equipment inspection images," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, Jul. 2024, doi: 10.1109/TIM.2024.3427802.

[6] W. Sun, K. Wei, Z. Liu, Q. Li, and X. Xu, "Linear quadratic gaussian control for wireless communication reliability for a mobile monitoring robot in a UHV power substation," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4149–4159, Sep. 2022.

[7] R. Sivapuram and R. Picelli, "Topology optimization of binary structures using integer linear programming," *Finite Elements Anal. Design*, vol. 139, pp. 49–61, Feb. 2018.

[8] L. Zhang, Y. Zhang, and F. van Keulen, "Topology optimization of geometrically nonlinear structures using reduced-order modeling," *Comput. Methods Appl. Mech. Eng.*, vol. 416, Nov. 2023, Art. no. 116371.

[9] X. Fu, P. Pace, G. Aloi, L. Yang, and G. Fortino, "Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107327.

[10] D. A. Marenda, R. Muhammad, and N. R. Syambas, "Ring topology optimization for wireless sensor network: A new heuristic method," *J. Commun.*, vol. 13, no. 8, pp. 463–467, 2018.

[11] C. Wang, N. Huang, Y. Bai, and S. Zhang, "A method of network topology optimization design considering application process characteristic," *Modern Phys. Lett. B*, vol. 32, no. 7, 2018, Art. no. 1850091.

[12] T. Qiu, B. Li, W. Qu, E. Ahmed, and X. Wang, "TOSG: A topology optimization scheme with global small world for industrial heterogeneous Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3174–3184, Jun. 2019.

[13] G. Prasad, D. Mishra, and R. H. Laskar, "Optimal new node insertion for strong minimum energy topology in IoT networks," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2021, pp. 1–4.

[14] S. Zhang, B. Yin, W. Zhang, and Y. Cheng, "Topology aware deep learning for wireless network optimization," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9791–9805, Nov. 2022.

[15] P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 103–118, Jan. 2024.

[16] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[17] Z. Li et al., "Network topology optimization via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2847–2859, May 2023.

[18] Z. Zhao, C. Liu, X. Guang, and K. Li, "A transmission-reliable topology control framework based on deep reinforcement learning for UWSNs," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 13317–13332, Aug. 2023.

[19] F. B. Mismar, J. Choi, and B. L. Evans, "A framework for automated cellular network tuning with reinforcement learning," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7152–7167, Oct. 2019.

[20] O. Franek, "Phasor alternatives to friis' transmission equation," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, pp. 90–93, 2018.

[21] A. Lazar, "The throughput time delay function of anM/M/1queue (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 914–918, Nov. 1983.

[22] D. Van Leemput, J. Bauwens, R. Elsas, J. Hoebeke, W. Joseph, and E. De Poorter, "Adaptive multi-PHY IEEE802.15.4 TSCH in sub-GHz industrial wireless networks," *Ad Hoc Netw.*, vol. 111, Feb. 2021, Art. no. 102330.

[23] C. Zhao, W. Sun, Z. Fang, J. Wang, Q. Li, and H. Zhang, "End-to-end delay optimisation for IEEE 802.11 string topology multi-hop wireless networks in overhead transmission line system," *IET Commun.*, vol. 15, no. 3, pp. 487–495, 2021.

[24] P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas, "Performance analysis of IEEE 802.11 DCF in presence of transmission errors," in *Proc. IEEE Int. Conf. Commun.*, 2004, pp. 3854–3858.

[25] I. Tinnirello, G. Bianchi, and Y. Xiao, "Refinements on IEEE 802.11 distributed coordination function modeling approaches," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1055–1067, Mar. 2010.

[26] Y. Xiao, "A simple and effective priority scheme for IEEE 802.11," *IEEE Commun. Lett.*, vol. 7, no. 2, pp. 70–72, Feb. 2003.

[27] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless lans," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1506–1515, Jul. 2005.

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

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[30] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[31] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," U.S. Patent 10 776 692, Sep. 15, 2020.

[32] N. Bof, R. Carli, and L. Schenato, "Lyapunov theory for discrete time systems," 2018, *arXiv:1809.05289*.

[33] Y. Shi, G. Qu, S. Low, A. Anandkumar, and A. Wierman, "Stability constrained reinforcement learning for real-time voltage control," in *Proc. Amer. Control Conf. (ACC)*, 2022, pp. 2715–2721.

[34] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[35] W. Sun et al., "Multi-agent reinforcement learning for dynamic topology optimization of mesh wireless networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 10501–10513, Sep. 2024, doi: 10.1109/TWC.2024.3372694.

[36] X. Liu, X. Han, N. Zhang, and Q. Liu, "Certified monotonic neural networks," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, 2020, pp. 15427–15438.

[37] S. Janković and M. Merkle, "A mean value theorem for systems of integrals," *J. Math. Anal. Appl.*, vol. 342, no. 1, pp. 334–339, 2008.
[38] C. Briat, "Convergence and equivalence results for the Jensen's inequality—Application to time-delay and sampled-data systems," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1660–1665, Jul. 2011.
[39] M. Darqaoui, M. Coulibaly, and A. Errami, "Cellular-V2X and VANET(DSRC) based end-to-end guidance for smart parking," in *Proc. 16th Int. Wireless Internet Conf.*, 2024, pp. 3–13.
[40] C. Lusty, V. Estivill-Castro, and R. Hexel, "TTWiFi: Time-triggered WiFi for mobile robotics in human environments," in *Proc. 16th Int. Wireless Internet Conf.*, 2023, pp. 14–28.
[41] A. I. Ameur, O. S. Oubbati, A. Lakas, A. Rachedi, and M. B. Yagoubi, "Efficient vehicular data sharing using aerial P2P backbone," *IEEE Trans. Intell. Veh.*, early access, Jun. 13, 2024, doi: 10.1109/TIV.2024.3414140.
[42] C. Dutriez, O. S. Oubbati, C. Gueguen, and A. Rachedi, "Energy efficiency relaying election mechanism for 5G Internet of Things: A deep reinforcement learning technique," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2024, pp. 1–6.
[43] O. S. Oubbati, H. Badis, A. Rachedi, A. Lakas, and P. Lorenz, "Multi-UAV assisted network coverage optimization for rescue operations using reinforcement learning," in *Proc. IEEE 20th Consum. Commun. Netw. Conf. (CCNC)*, 2023, pp. 1003–1008.
[44] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.
[45] X.-L. Huang, X. Ma, and F. Hu, "Machine learning and intelligent communications," *Mobile Netw. Appl.*, vol. 23, pp. 68–70, Feb. 2018.
[46] X.-L. Huang, X. Tang, X. Huan, P. Wang, and J. Wu, "Improved KMV-cast with BM3D denoising," *Mobile Netw. Appl.*, vol. 23, pp. 100–107, Feb. 2018.

Qingwei Tang is currently pursuing the Ph.D. degree with the School of Electrical and Automation Engineering, Hefei University of Technology. His research primarily explores the application of multi-agent reinforcement learning in complex systems, including wireless communication networks, modern power systems, and smart grids. His work aims to advance the integration of intelligent systems within critical infrastructure, promoting more sustainable, and resilient technological innovations.

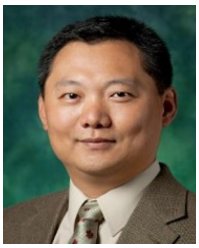
Wei Sun (Senior Member, IEEE) received the B.E. degree in automation, the M.S. degree in detection technology and automatic equipment, and the Ph.D. degree in electrical engineering from the Hefei University of Technology, China, in 2004, 2007, and 2012, respectively, where he is currently a Professor. His research interests include wireless networks, networked control systems, and microgrids.

Zhi Liu (Senior Member, IEEE) received the Ph.D. degree in informatics from the National Institute of Informatics. He is currently an Associate Professor with The University of Electro-Communications. His research interest includes video network transmission and mobile edge computing. He is currently an Editorial Board Member of *Wireless Networks* (Springer) and IEEE TRANSACTIONS ON MULTIMEDIA.

Yang Xiao (Fellow, IEEE) received the B.S. and first M.S. degrees in computational mathematics from Jilin University, Changchun, China, in 1989 and 1991, respectively, and the second M.S. and Ph.D. degrees in computer science and engineering from Wright State University, Dayton, OH, USA, in 2000 and 2001, respectively. He is currently a Full Professor with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA.

He directed more than 20 doctoral dissertations and supervised over 20 M.S. theses/projects. He has authored or co-authored more than 300 Science Citation Index (SCI)-indexed journal papers (including over 70 IEEE/ACM Transactions) and 300 Engineering Index (EI)-indexed refereed conference papers and book chapters related to these research areas. His research interests include cyber-physical systems, the Internet of Things, security, wireless networks, smart grids, and telemedicine. He was the recipient of the IEEE TNSE Excellent Editor Award in 2022 and 2023. He was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004, involving the IEEE 802.11 (Wi-Fi) standardization work. He was a Guest Editor 37 times for different international journals, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS from 2022 to 2023, IEEE Transactions on Network Science and Engineering in 2021, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING in 2021, IEEE NETWORK in 2007, IEEE WIRELESS COMMUNICATIONS in 2006 and 2021, IEEE Communications Standards Magazine in 2021, and *Mobile Networks and Applications* (ACM/Springer) in 2008. He is also the Editor-in-Chief of *Cyber-Physical Systems*, *International Journal of Sensor Networks*, and *International Journal of Security and Networks*. He has been an Editorial Board Member or an Associate Editor for 20 international journals, including the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING since 2022, IEEE TRANSACTIONS ON CYBERNETICS since 2020, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2015, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2009, and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS from 2007 to 2014. He is/was a member of technical program committee for more than 300 conferences. He is a Fellow of IET, AAIA, AIIA, and ACIS.

Qiyue Li (Senior Member, IEEE) received the B.E. degree in electronic engineering from Wuhan University, Wuhan, China, in 2003, and the Ph.D. degree in communication and information system from the University of Science and Technology of China, Hefei, Anhui, China, in 2008. From 2008 to 2011, he was a Postdoctoral Researcher with the School of Computer Science and Technology, University of Science and Technology of China. He is currently a Professor with the Hefei University of Technology, Hefei. His research interests include wireless networks and indoor localization using wireless networks.



Xiaohui Yuan (Senior Member, IEEE) is an Associate Professor and the Director of the Computer Vision and Intelligent Systems Lab, University of North Texas, Denton, TX, USA. His research interests include artificial intelligence and machine learning. He was a recipient of the Ralph E. Powe Professor Award in 2008 and the U.S. Air Force Visiting Professor Award in 2011, 2012, and 2013, respectively. He serves as an associate editor, an editorial board member, and a guest editor for several journals, and an organizing member for many international conferences.

Qian Zhang is an Experimental Teacher with the Hefei University of Technology, specializing in image processing, pattern recognition, and artificial intelligence.