Energy Consumption Optimization of UAV-Assisted Traffic Monitoring Scheme With Tiny Reinforcement Learning

Xiangjie Kong, *Senior Member, IEEE*, Chenhao Ni, Gaohui Duan, Guojiang Shen, Yao Yang, and Sajal K. Das, *Fellow, IEEE*

***Abstract*—Unmanned aerial vehicles (UAVs) equipped with high-definition cameras have the capability to capture comprehensive and multiangled images of road conditions, facilitating more efficient collection of pertinent road data. However, drones encounter challenges in performing related tasks for an extended period due to their limited energy capacity. Therefore, a crucial concern is how to plan the path of UAVs and minimize energy consumption. To address this problem, we propose a multiagent deep deterministic policy gradient (MADDPG)-based algorithm for UAV path planning (MAUP). Considering the energy consumption and memory usage of MAUP, we have conducted optimizations to reduce consumption on both fronts. First, we define an optimization problem aimed at reducing UAV energy consumption. Second, we transform the defined optimization problem into a reinforcement learning problem and design MAUP to solve it. Finally, we optimize energy consumption and memory usage by reducing the number of neurons in the hidden layer of MAUP and conducting fine-grained pruning on connections. The final simulation results demonstrate that our method effectively reduces the energy consumption of UAVs compared to other methods.**

***Index Terms*—Energy consumption optimization, multiagent deep deterministic policy gradient (MADDPG), multiagent reinforcement learning (MARL), tiny machine learning (Tiny ML), unmanned aerial vehicle (UAV) path planning.**

I. Introduction

DUE THE ongoing developments and miniaturization of electronic systems, the proliferation of Internet of Things (IoT) devices in real world has exhibited an exponential growth. These devices generate vast amounts of data, which is subsequently processed using machine learning algorithms to extract valuable information. The escalating demand for effectively managing the overwhelming volume of data generated by IoT devices, coupled with the growing expectation for enhanced responsiveness in such systems, has prompted the migration of data processing in cloud computing toward edge computing even extending to the devices themselves. These systems can be supported by machine learning algorithms, particularly reinforcement learning, which facilitates interaction with the environment and the accumulation of rewards based on executed actions. However, given the constraints in computational resources, storage space, and energy in IoT and edge devices, the emergence of tiny machine learning (Tiny ML) has become inevitable [1]. Tiny ML represents a rapidly evolving domain encompassing machine learning technologies and applications, including algorithms, hardware, and software, designed to facilitate on-device sensor analytics at ultralow power [2].

By harnessing the capabilities of Tiny ML, IoT devices can demonstrate an increasingly diverse range of functionalities. However, their inherent lack of mobility poses limitations on traffic monitoring tasks. In such scenarios, the integration of unmanned aerial vehicles (UAVs) presents a compelling solution. With their exceptional maneuverability, UAVs can dynamically select surveillance areas and effectively cover a significantly larger monitoring range compared to conventional methods. Meanwhile, the monitoring of traffic conditions holds paramount importance in smart cities, intelligent transportation systems, and other related domains [3], [4]. Due to the diversity of UAV types, UAVs can adapt to various task requirements. Moreover, these UAVs can also accommodate various camera devices, enabling footage capture from multiple perspectives [5], [6]. In the event of an emergency, monitoring traffic conditions can provide timely warnings to drivers and effectively mitigate the occurrence of traffic accidents, thereby reducing both human casualties and property damage [7], [8]. The integration of Tiny ML in the UAV domain can enhance the capabilities of small drones, such as improving robustness during hovering and way finding. Concurrently, this facilitates small UAVs to demonstrate heightened intelligence through locally processing rather than transmitting data to a base station (BS), thereby augmenting real-time performance and privacy for tasks.

Manuscript received 28 November 2023; revised 20 January 2024; accepted 5 February 2024. Date of publication 13 February 2024; date of current version 7 June 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4501500 and Grant 2022YFB4501504; in part by the National Natural Science Foundation of China under Grant 62072409 and Grant 62073295; and in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003. The work of Sajal K. Das was supported by the US National Science Foundation under Grant CNS-2008878, Grant EPCN-2319995, Grant OAC-2104078, and Award SCC-1952045. *(Corresponding author: Guojiang Shen.)*

Xiangjie Kong, Chenhao Ni, and Guojiang Shen are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xjkong@ieee.org; qnch_ut@outlook.com; gjshen1975@zjut.edu.cn).

Gaohui Duan is with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: dghhuihui@gmail.com).

Yao Yang is with the Research Center for Space-Based Computing System, Zhejiang Laboratory, Hangzhou 311100, China (e-mail: yangyao@zhejianglab.com).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

Digital Object Identifier 10.1109/JIOT.2024.3365293

However, the rational planning of UAV routes remains a significant challenge due to their limited energy capacity [9]. The primary objective of route planning for UAVs is to ascertain the optimal collision-free trajectory that can effectively accomplish the desired outcome while simultaneously satisfying criteria pertaining to distance, cost, time, and other pertinent factors. The accomplishment of these objectives necessitates the incorporation of various constraints pertaining to the physical attributes of the UAV, such as energy and velocity, into path planning. To optimize UAV energy consumption, it is imperative to devise a judicious path planning strategy. Considering the aforementioned information, the primary focus of this article lies in devising an efficient path for the UAV to minimize its energy utilization during monitoring operations.

Several viable solutions have been proposed in existing studies to address the problem of UAV path planning. However, these studies utilize traditional path planning methods and heuristics, both of which rely on prior knowledge. The traditional methods often encounter local optimum situations. During the phase of UAVs performing traffic monitoring tasks, the UAV's state undergoes constant changes. Hence, real-time decisions are imperative to determine the flight plan of the UAV at each time point.

Deep reinforcement learning (DRL) methods have been applied in many scenarios due to their powerful real-time decision-making capabilities [10], [11], [12], [13]. While traffic monitoring scenarios are designed with multiple UAVs, each UAV also needs to adjust its flight strategy based on the flight strategies of other UAVs during their flights. Traditional single-agent reinforcement learning methods involve only one agent, and the actions taken are solely related to this intelligence in order to learn a control strategy. If a single-agent reinforcement learning method is employed and UAVs are represented as a unified agent, the action space will expand exponentially with an increasing number of UAVs, resulting in the challenge of dimensional explosion. In addition, if each UAV is considered as an agent and trained separately using single-agent reinforcement learning methods, it is easy to ignore the flight strategies of other UAVs. In the field of reinforcement learning, multiagent reinforcement learning (MARL) methods can compensate for the shortcomings of single-agent reinforcement learning methods.

This article presents a comprehensive approach to optimize the energy consumption of UAVs by considering both hovering and flight operations in an integrated manner. The optimization problem is then transformed into a reinforcement learning problem. The energy consumption of UAVs extends beyond flight operations and encompasses energy usage in communication and computation processes. Due to the decentralized execution and centralized training paradigm of multiagent deep deterministic policy gradient (MADDPG)-based algorithm, interagent communication is unnecessary. Agents solely make predictions about other agents, thereby the algorithm reducing communication energy consumption among them. This article designs a MADDPG-based algorithm for UAV path planning (MAUP). Furthermore, in order to deploy smaller UAVs and save more energy, we reduce the size of the MAUP by modify the hidden layer neurons and perform fine-grained pruning on connections, transforming original MAUP into Tiny MAUP. The primary contributions of this article are listed below.

1) We formulate an optimization problem to minimize UAVs energy consumption in a highway road monitoring scenario. For this problem, we consider the mobility capability of the UAVs, as well as their energy consumption during flight, hovering and communication.
2) We design a MADDPG-based algorithm for UAV path planning to solve the optimization problem. MADDPG is more suitable in the energy-saving scenario of UAVs due to its feature: centralized training and decentralized execution, which eliminate the need for communication among UAVs before making decisions.
3) We propose the Tiny MAUP algorithm, which aims to reduce the energy loss caused by the original algorithm. It is modified from the original MAUP algorithm during hidden layer node modify and pruning. Compared to the original MAUP, Tiny MAUP requires less computation and memory usage.

The remainder of this article is organized as follows. Section II displays the related work. Section III shows the system model. In Section IV introduces the problem formulation and proposed MADDPG-based UAV path planning algorithm. The simulation experiment settings and experimental findings are thoroughly described in Section V. Section VI provides the conclusion.

II. Related Work

This section presents a comprehensive overview of research on UAVs-based traffic monitoring methods, DRL algorithms for drones' control, and Tiny ML-based control for drones.

A. Monitoring Traffic Conditions With UAV

Khan et al. [14] proposed an incognito airborne traffic surveillance system based on UAVs utilizing 5G technology. This system leverages the capabilities of UAVs and 5G to effectively monitor, track, and regulate speed as well as detect any illicit traffic behavior or suspicious vehicles on highways and roads. Lyu et al. [15] proposed a multiobjective optimization problem that aims to maximize data collection and energy transfer while minimizing UAV energy consumption during the UAV serving time. Then they use multiobjective joint optimization oriented DDPG algorithm (MJDDPG)-based recourse allocation algorithm to solve this problem. A proactive energy-efficient and reliable collaborative scheme between UAVs and VANETs is presented in [16]. The authors proposed an innovative proactive approach to address the challenges posed by highly mobile UAV networks. The primary focus of this method is to establish a reliable and energy-efficient routing mechanism for UAV systems. The reliance of UAV-based architectures on terrestrial networks is hindered by exorbitant deployment costs. To overcome

this limitation, Bashir et al. [17] proposed a novel closed-loop control architecture for highway traffic surveillance that enhances effectiveness and adapts to varying traffic patterns.

B. Reinforcement Learning Methods for UAV Path Planning

Huang et al. [18] applied deep Q-network (DQN) to UAV navigation, utilizing DQN to search for optimal flight strategies. Zhang et al. [19] proposed a deep-constrained Q algorithm that formulates the problem of 3-D dynamic motion of UAVs under coverage constraints as a Markov decision process (MDP). They then utilized prior knowledge in DQN to eliminate ineffective actions, thereby finding better flight paths. Liu et al. [20] investigated problems with the use of UAVs for task offloading employing mobile edge computing and created an algorithm based on DDQN approach to maximize the overall throughput. Bayerlein et al. [21] designed a novel RL method for obtaining data from IoT devices using UAVs. They leveraged DDQN to strike the right balance of data collection, obstacle avoidance and mission time minimization.The proposed multi-UAV trajectory optimization algorithm by Ning et al. [22] is based on partial information and allows for distributed execution of flying actions. To the best of our knowledge, this is the first work to achieve distributed control of multi-UAV trajectories in scenarios with probabilistic time-varying service preferences. Although numerous studies have explored differentiated services offered by service providers, their applicability to UAV-based networks is limited due to the unique characteristics of these networks. To the best of our knowledge, Wang et al. [23] are the pioneers in investigating differentiated services with distinct service providers in UAV-based networks.

C. Tiny Reinforcement Learning for UAV control

The control of UAV, can be classified into low-level and high-level control. Low-level control focuses on achieving specific velocity or position objectives, while high-level control involves determining the subsequent destination. In the forthcoming sections, we will present two distinct approaches' related work for UAV control.

1) Low-Level Control: Lambert et al. [24] proposed a model-based reinforcement learning method for quadrotor hovering, which is suitable for dynamic systems with unknown priors and more applicable to real-world scenarios. Molchanov et al. [25] designed a low-level control approach for hovering based on PPO, which replaces the normal PID method and achieves more robust evaluation. They deploy it on three different quadrotors to demonstrate its effectiveness.

2) High-Level Control: Duisterhof et al. [26] applied DQN for a high-level control algorithm, which can be deployed on the nano quadrotor to seek light and avoid obstacles. Ho et al. [27] proposed a method based on trust region policy optimization (TRPO) to solve the nonconvex problem of wireless service provisioning through a quadcopter in a dynamic environment with continuous action space. Kang et al. [28] integrated a substantial volume of simulated data with a limited amount of real-world experience to train DRL-based autonomous flight collision avoidance strategies. This approach is rooted in comprehending the physical characteristics and dynamics of vehicles in the real world, while simultaneously acquiring visual in-variance and patterns from simulations.

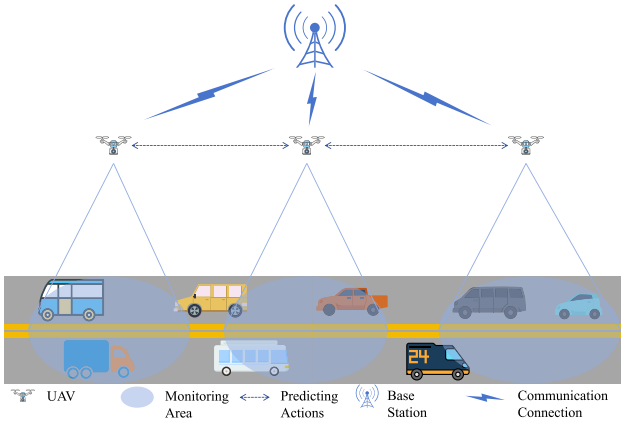


Fig. 1. UAV assisted traffic monitoring model.

III. System Model

The scenario investigated in this study pertains to the UAV surveillance of the highway road. A square area with side length S^l is considered as the area where the UAV performs the monitoring task. In this scenario, several UAVs are deployed on the highway, and the collection of UAVs is represented as $U = 1, 2, \ldots, M$, the sensing module and camera are integrated into each UAV. There are moving vehicles on the highway and the UAVs can collect relevant data (e.g., vehicle speed) within their coverage area. The data collected by the UAVs is transmitted to a BS where a control system is deployed to analyze the data collected by the UAVs. The energy consumption of the UAV is related to the flight path of the UAV. This article assumes that the UAV flies at a fixed altitude. The longest distance a UAV can fly at once is $d_{\max}$, the total execution time of the monitoring task is t_{total}. The execution time of UAV in the whole system is divided into discrete time slots, and the duration of each time slot is τ, where $\tau = t_{\text{total}}/T$. In each time slot, UAV performs road monitoring tasks by flying or hovering. In slot t, the horizontal position of the UAV n is denoted by $U_n = [x_n(t), y_n(t)]$. The model diagram is shown in Fig. 1.

A. Communication Model

The UAV needs to transmit the collected road traffic information to the BS in time. Assume that the flight height of UAV is fixed, and the set of horizontal positions of each UAV is $D = \{U_1, U_2, \ldots, U_M\}$ within the time slot t. The horizontal positions of BS is $U_B = [x_B, y_B]$. In time slot t, the distance between the UAV and the BS is represented by $d_j = \sqrt{\|U_j - U_B\|^2 + H^2}$, where the perceived data is transmitted through a wireless channel. Assume that a Line-of-Sight (LoS) connection is established between the UAV and the BS.

The channel gain between the BS and the UAV is denoted by $G_j = G_j^r G_j^t$, where G_j^r and G_j^t represents the gain of receiving antenna and gain of transmitting antenna, respectively. In the LoS connection scenario [29], the received power can be expressed as

$$P_j^r = P_j^t G_j \left(\frac{\lambda}{4\pi d}\right)^{\alpha} \tag{1}$$

where P_j^t is the transmitted signal power, λ is the wavelength, and d is the distance between the BS and UAV. In the scenario of free-space path loss, a path loss exponent α is set to 2. The signal-to-interference-plus-noise ratio (SINR) between UAV and BS is defined as [30]

$$\gamma_j = \frac{P_j^r}{\sigma^2 + \sum_{i \in D \setminus j} P_i^r} \tag{2}$$

where σ^2 is the white Gaussian noise and the second term in the denominator represents channel interference from other UAVs. Based on Shannon theory, the data rate between the UAV j and the BS is expressed as

$$R_j(t) = b_j(t) \log_2\left(1 + \gamma_j\right). \tag{3}$$

B. UAV Energy Consumption Model

The energy consumption of UAVs generally consists of two parts: 1) communication energy consumption and 2) flight hover energy consumption. Since the UAV needs to perform task monitoring in the air, it will generate flight energy consumption. Flight energy consumption is usually determined by the speed and acceleration of the UAV. In this article, the flight hover energy consumption of UAV is mainly considered. Based on [31], the flight power consumption of UAV is modeled as

$$\begin{aligned} P(V) = {} & P_0\left(1 + \frac{3V^2}{U_{\text{tip}}^2}\right) \\ & + P_i\left(\sqrt{1 + \frac{V^4}{4v_0^4}} - \frac{V^2}{2v_0^2}\right)^{\frac{1}{2}} \\ & + \frac{1}{2} d_0 \rho s A V^3 \end{aligned} \tag{4}$$

where P_0 is the blade profile power, P_i is the induced power, V is the speed, U_{tip} represents the rotor blade speed, and v_0 is known as the mean rotor induced velocity in hover. ρ, s, A, d_0 represent air density, rotor solidity, rotor disk area and fuselage drag ratio, respectively. When the flight speed of the UAV is 0, it enters a state of hover. The hover power of the UAV is $P(0) = P_0 + P_i$, where P_0 and P_i are a finite value determined by the weight of the UAV. They represent the air density and the area of the rotor disc, respectively. In time slot t, each UAV takes a decision that consists of two parts: 1) flight direction $\omega(t)$ and 2) distance d_t, where $\omega(t) \in [0, 2\pi], d_t \in [0, d_{\max}]$. Therefore, in time slot $t+1$, the horizontal position coordinate of the UAV is $U_n(t+1) = [x_n(t) + d_t \cos(\omega(t)), y_n(t) + d_t \sin(\omega(t))]$. The flight path of each UAV must be reasonable when flying. In time slot t, the flight time and hover time of UAV are $T_f(t)$ and $T_h(t)$, respectively. Then the energy consumption of the UAV during the time slot t can be expressed as

$$E_j(t) = P(V) T_f(t) + P(0) T_h(t). \tag{5}$$

Hence, the overall flight and hovering energy consumption of UAV j in the execution of traffic monitoring tasks is expressed as

$$E_j = \sum_{t=1}^{T} E_j(t) \tag{6}$$

where T represents the upper bound of the time slot t.

IV. Multiagent DRL Method for UAV Path Planning

This section shows a problem formulate, overview of MARL and provides further analysis of the energy consumption minimization problem formulated in the previous section. By analyzing the system model, the energy optimization problem is transformed into a reinforcement learning problem. A UAV path planning algorithm based on MADDPG is designed to search for the optimal flight strategy, ensuring that the UAV can successfully accomplish monitoring tasks while reducing the energy consumption resulting from the execution of these tasks.

A. Problem Formulation

Based on the discussion in Section III, the final optimization problem is set as the minimization of energy consumption during the flight and hovering of UAV. However, during the UAV flight process, the flight path must be reasonable and satisfy certain constraints. The main constraints of the system model can be summarized as follows:

$$0 \le d_t \le d_{\max} \tag{7}$$

$$0 \le \omega(t) \le 2\pi \tag{8}$$

$$x_n(t), y_n(t) \in \left[0, S^l\right] \tag{9}$$

$$0 \le T_f(t) \le \tau \tag{10}$$

$$0 \le T_h(t) \le \tau \tag{11}$$

where (7) and (8) stipulate that the flight direction and flight distance of the UAV are within the interval $[0, 2\pi]$ and $[0, d_{\max}]$, respectively. Constraint (9) guarantees that each UAV is within the designated area. Constraints (10) and (11) put certain limits on the flight and hovering time of each UAV in each time slot, respectively, i.e., the flight and hovering time of the UAV is at least 0 and cannot exceed the size of each time slot. According to the above analysis, the final optimization problem is defined as follows:

$$\begin{aligned} \min \ & \sum_{j=1}^{M} E_j \\ \text{s.t.} \ & \text{Constraints} \quad (7)-(11). \end{aligned} \tag{12}$$

B. MARL—An Overview

MARL involves a set of agents in a sequential decision problems [32], [33]. When there are multiple agents interacting with the environment at the same time, the whole system becomes a multiagent system. From a more intuitive perspective, although each agent's ultimate goal remains to maximize its own reward, the received reward is no longer solely determined by its individual actions but also influenced by the collective actions of all agents. To optimize long-term payoff, each agent must consider the strategies employed by other agents.

When sequential decision making is extended to multiple agents, Markov games (MG) provide a theoretical framework [34], [35], [36] that was originally introduced by Littma [37] to extend MDP to multiple agents that interact with the environment simultaneously and also interact with each other informally. Let $N > 1$ denote the number of agents and S be the set of states observed by all the agent, and the joint action space of all the agent is denoted as $A = A_1 \times A_2 \ldots A_N$.

In state S, each agent chooses an action a_i according to its respective policy, and the joint action $\underline{a} = [a_i]_{i \in N}$ will be executed in the environment. Based on the state transfer function $P : S \times A \times S \rightarrow [0, 1]$, the environment state changes from S to S'. Each agent gets an instantaneous reward r_i according to the reward function $R_i : S \times A \times S \rightarrow R$. The state transfer and reward functions in MG depend on the joint action space where each agent aims to find the optimal strategy that maximizes long-term returns.

C. Problem Transformation

Since UAVs are in constant motion, at each time slot, each UAV needs to determine its flight strategy based on its current state. In other words, the position information of the UAVs is continuously changing. Therefore, the UAV path planning problem can be viewed as a sequential decision-making problem. Traditional sequential decision-making algorithms, such as dynamic programming, search algorithms, and heuristic algorithms, can solve such problems, but they often come with high-computational costs or complexities. In comparison, reinforcement learning algorithms can make decisions based on the current state of the system in the UAV-assisted traffic monitoring network. During the training process, reinforcement learning algorithms continuously optimize the UAV path planning strategy based on the rewards obtained from interacting with the environment, ultimately finding the optimal path planning solution. Hence, we transform the optimization problem in (12) into a DRL problem.

However, traditional single-agent reinforcement learning models have only one agent and are prone to issues like dimensional explosion. The scenario considered in this article involves multiple UAVs, where each UAV needs to consider the flight strategies of other UAVs in order to cooperate and accomplish monitoring tasks. Traditional single-agent reinforcement learning methods do not consider these factors when selecting strategies since they involve only one agent. Therefore, single-agent reinforcement learning methods are not well-suited for the application scenario discussed in this article. Furthermore, according to (7) and (8), the flight direction and distance of UAVs are continuous values. Hence, traditional reinforcement learning methods used for discrete resource allocation are not suitable for this application scenario. On the other hand, the MADDPG method, designed for MARL with continuous action spaces. Which is particularly well-suited for the UAV traffic monitoring scenario.

Based on these two points, this article presents a method based on MADDPG to select the optimal UAV path planning solution. The algorithm framework is illustrated in Fig. 2, and the following sections explain the three key components that need to be constructed when designing this algorithm. In this figure, θ and φ represent the parameters of actor network and critic network, respectively. μ_θ represents the policy generated by the agent network. And $Q_{M\varphi}$ represents the Q value generated by the critic network.

1) *State:* The status information of each UAV contains two parts: a) its own location information and b) the location information of other drones.
2) *Action:* This article is based on the assumption that all UAVs have the same action space. The actions that can be taken by UAVs in each time slot consist of two components: a) the flight direction $\omega(t) \in [0, 2\pi]$ and b) the flight distance $d_t \in [0.d_{\max}]$. After each flight, the position information of the UAVs will change.
3) *Reward:* In the road traffic monitoring scenario, each UAV wants to minimize its energy consumption, the main objective of this article is to minimize the total energy consumption of the UAV, but the goal of the reinforcement learning method is to maximize the reward, in addition, during the UAV training process, there will be situations where the task area is exceeded and some penalties need to be imposed, therefore, the reward function R_i is set as follows:

$$R_i = \begin{cases} P_1, & \text{If UAV flies away from the task area} \\ \frac{1}{E_j(t)}, & \text{Otherwise.} \end{cases} \tag{13}$$

D. MADDPG-Based UAV Path Planning Algorithm

Based on the design of system states, actions, and reward function, this article presents a MADDPG-based UAV path planning algorithm. The pseudo-code for this algorithm can be found in Algorithm 1. The main steps of the algorithm are as follows.

First, the parameters of the MADDPG network are initialized. Similar to the deep deterministic policy gradient (DDPG) network structure, the MADDPG network consists of actor and critic networks, along with their corresponding target networks. Initially, the target network parameters of the actor and critic networks are set to be the same as the parameters of the actor and critic networks. Since the experience replay buffer is empty at the beginning, it is in an empty state. For each UAV, the initial own state is observed, and the state set is represented as $S = \{o_1, o_2, \ldots, o_M\}$. Under the system state, each UAV selects an action based on the existing policy and noise, the action set is expressed by $a = \{a_1, a_2, \ldots, a_M\}$. This action determines the flight direction and distance for each UAV. Subsequently, the UAVs execute their respective

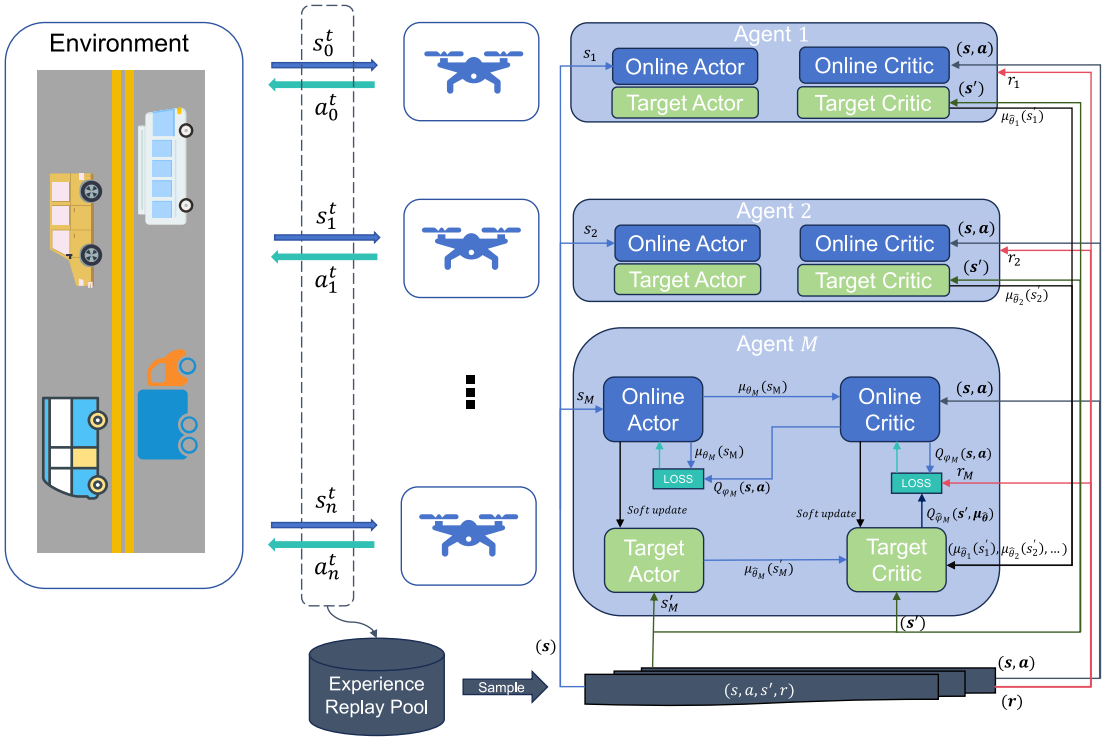


Fig. 2. MAUP framework.

actions, and after taking these actions, each UAV receives an immediate reward. The set of rewards is denoted as r. At the same time, the set of system states observed by each UAV changes from S to S'.

Then, the tuple (S, a, r, S') is put in the experience replay buffer. For each UAV, k samples are sampled from the experience replay buffer so that the target value is calculated by the Critic target network of each UAV. The target network of UAV j calculates the target value based on the sample with the following equation:

$$y_j^i = r_j^i + \gamma Q_j'(s'^i, a_1'^i, \ldots, a_M'^i)\Big|_{a_j'^i = \mathcal{L}'\left(o_j'^i | \theta_j^{\mathcal{C}'}\right)} \tag{14}$$

where $Q_j'(s'^i, a_1'^i, \ldots, a_M'^i)$ is a centralized action-value function that takes as input the actions of all agents, $a_1'^i, \ldots, a_M'^i$, in addition to some state information s', and γ is the discount factor, and r_j^i is the reward for performing action a_j^i. Then, the critic network is updated by minimizing the result obtained from the loss function. The loss function is expressed as follows:

$$\mathcal{L}\left(\theta_j^Q\right) = \frac{1}{k}\sum_{i=1}^{k}\left(Q_j(S^i, a^i) - y_j^i\right)^2. \tag{15}$$

Next, updating actor networks with policy gradients

$$\begin{aligned}\nabla_{\theta_j^{\mathcal{C}}} J\left(\theta_j^{\mathcal{L}}\right) &= E\nabla_{\theta_j^{\mathcal{L}}}\mathcal{L}\left(o_j \mid \theta_j^{\mathcal{L}}\right) \\ &\quad \nabla_a Q_j(S, a_1, \ldots, a_M)|_{a_j = \mathcal{L}o_j|\theta_j^{\mathcal{L}}} \\ &= \frac{1}{k}\sum_{i=1}^{L}\nabla_{\theta_j^{\mathcal{L}}}\mathcal{L}(o_j^i \mid \theta_j^{\mathcal{L}}) \\ &\quad \nabla_a Q_j(S^i, a_1^i, \ldots, a_M^i)|_{a_j^i = \mathcal{L}(o_j^i|\theta_j^u)}.\end{aligned} \tag{16}$$

Finally, the target networks are updated by leveraging soft updating method

$$\begin{aligned}\theta_i^{\mathcal{L}'} &\leftarrow \tau\theta_i^{\mathcal{L}} + (1-\tau)\theta_i^{\mathcal{L}'} \\ \theta_i^{Q'} &\leftarrow \tau\theta_i^{Q} + (1-\tau)\theta_i^{Q'}\end{aligned} \tag{17}$$

where τ is the soft update coefficient.

E. Algorithm Analyses

The algorithm incorporates two-layer multilayer perceptions (MLPs). In each epoch, the time complexity of the algorithm can be represented by the following equation:

$$O = O\left(\sum_{i=1}^{l+1} L_{i-1}L_i + 2L_i\right) \tag{18}$$

where L_i represents number of the i layer's neurons, L_0, L_{l+1} represent dimensions of input data and dimension of output data, and l represents number of hidden layer neurons. Based on (18), the time complexity of actor network can be represented as

$$\begin{aligned}O_a &= O(L_S H + 2H + H^2 + 2H + HL_A + 2L_A) \\ &= O(4H + 2L_A + H(L_S + L_A) + H^2)\end{aligned} \tag{19}$$

where H represents the number of neurons in the hidden layer, and L_S and L_A are state dimension and actor dimension, respectively. And time complexity of critic network can be represented as

$$\begin{aligned}O_c &= O((L_S + L_A)H + 2H + H^2 + 2H + H) \\ &= O((L_S + L_A)H + 2H^2 + 5H).\end{aligned} \tag{20}$$

Because, the MAUP algorithm has target networks for actor and critic, and target actor and critic networks structure are

Algorithm 1 MADDPG-Based UAV Path Planning Algorithm

Initialize the parameters $\theta^{\mathcal{L}}$ and $\theta^{\mathcal{Q}}$ of actor network and critic network, and initialize the experience replay buff E to Empty.
Initialize target network parameters $\theta^{\mathcal{L}'}$ and $\theta^{\mathcal{Q}'}$ corresponding to the actor and critic network: $\theta^{\mathcal{L}'} \leftarrow \theta^{\mathcal{L}}, \theta^{\mathcal{Q}'} \leftarrow \theta^{\mathcal{Q}}$.
for episode=1,..., N **do**
    A random process is initialized for action exploration.
    All UAVs observe initial state $S = \{o_1, o_2, \ldots, o_M, \}$.
    for $t = 1,\ldots,$ T **do**
        Each UAV chooses action $a_m = \mathcal{L}(o_m|\theta_m^{\mathcal{L}}) + N_t$ based on available strategies and noise.
        Each UAV execute corresponding action $a = \{a_1, \ldots, a_M\}$, calculate instant reward. The set of reward is r. Meanwhile, the state is updated from S to S'.
        $R_i = R_i + r_t$.
        Store tuple(S, a, r, S') in E.
        for agent $i = 1, \ldots, M$ **do**
            Randomly sample a batch of K tuples (S^k, a^k, r^k, s'^k) from E.
            Get the target value y_t on the basis of (14).
            Update the critic network via minimizing the loss using (15).
            Update the actor network through (16).
        end for
        The target networks are updated via (17)
    end for
end for

same as actor and critic network. In each epoch, the algorithm time complexity can be represented as

$$O_{\text{all}} = 2MO_a + 2MO_c = O(4ML_A + 18MH + 4MH(L_S + L_A) + 4MH^2). \quad (21)$$

F. Tiny MAUP

Furthermore, we are striving to optimize computation and memory usage to facilitate the deployment of small drones and nano quadrotors. Additionally, the incorporation of Tiny MAUP in conventional UAVs holds promising potential for minimizing energy consumption and enhancing endurance. We will optimize the MAUP by adjusting the number of hidden layer neurons and eliminating unnecessary connections between neurons.

First, we will reduce the number of neurons in the hidden layers because the original MAUP algorithm had 128 neurons in the hidden layer, which is a large number for UAV memory and requires a lot of computations. And UAVs are involved in traffic monitoring scenarios, which may require a significant amount of memory space to store high-definition pictures or videos. Additionally, in Section IV-E, we analyze the time complexity of MAUP. Our analysis reveals a strong correlation between the time complexity and the number of hidden layer neurons. Reducing the number of hidden layer neurons not only decreases computational energy consumption but also memory usage can be reduced.

Second, given the limited number of neurons per layer, we will employ fine-grained pruning on connections. Our fine-grained pruning is based on the L1 norm pruning method, which prunes low-weight connections. At the same time, a low weight indicates that the connection is not important.

TABLE I
PARAMETER SETTING

Parameter	Value
Altitude of the UAV	50m
The longest distance a UAV can fly at once	40m
Size of each time slot	4s
Number of time slots	200
Blade profile power of UAV	158.76W
Induced power of UAV	88.63W
Mean rotor induced velocity in hover	4.03m/s
Fuselage drag ration	0.6
Rotor solidity	0.05
Air density	1.225 kg/m^3
Rotor disc area	0.503m^2
Tip speed of the rotor blade	120m/s

V. PERFORMANCE EVALUATION

A. Simulation Setup

In this article, it is assumed that the UAV performs traffic monitoring tasks on the highway, and the flight height of the UAV is a fixed value, and the height is set as 50 m. The maximum distance of a flight of the UAV is 40 m, and the flight speed of the UAV is 20 m/s. The induced power of UAV is 88.63w, blade profile power is 158.76w, blade speed is 120 m/s, and air density is 1.225 kg/m^3. Table I describes other environment parameters. An evaluation index are adopted in this article, namely, the total flight and hover energy consumption of all UAVs.

The MAUP algorithm runs on a windows 64-bit operating system with a Core i5-10400 processor. The neural network framework is PyTorch, which is version 1.10.2 and python version 3.6. MAUP related core parameters are shown in Table I.

To verify the effectiveness of MAUP, the bellowed two algorithms are adopted for comparison.

1) *Random:* Each UAV randomly chooses its action from the action space at each time slot to determine its flight direction and distance. If the action chosen by the UAV causes the UAV to exceed the boundary, the UAV will remain in its original position.
2) *DDPG:* DDPG is a typical DRL algorithm which can be utilized in the continuous scenario.

We do not compare MAUP with DQN-based algorithms and traditional optimization approaches because the former cannot handle problems with continuous action spaces due to the extremely large Q-table in such scenarios, while the latter are unable to cope with dynamic environments and the high complexity of real-time problems.

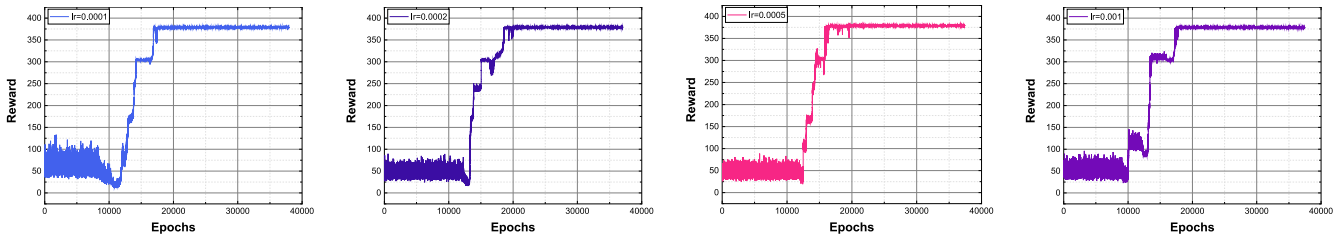


Fig. 3. Convergence performance of MAUP under different learning rates. (a) Learning rate 0.0001. (b) Learning rate 0.0002. (c) Learning rate is 0.0005. (d) Learning rate is 0.001.

TABLE II
EVALUATION IN DIFFERENT HIDDEN LAYER NEURONS

Hidden Layer Neurons	128	64	32	16
The mean of stable epochs in 5 experiments	9000	11000	15000	20000
learning rate	0.0005	0.0005	0.0005	0.0005
soft update rate	0.01	0.01	0.01	0.01
model size (16 neurons model size is 1)	11.89	3.79	1.63	1

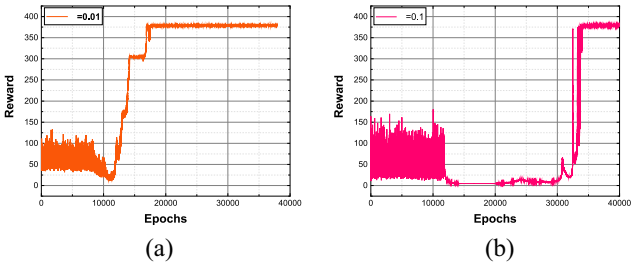


Fig. 4. Convergence performance of MAUP under different soft update coefficient. (a) Soft update coefficients is 0.01. (b) Soft update coefficients is 0.1.

B. Experimental Results

Table II presents training time and model size for same models with different numbers of hidden layer nodes. During the average convergence round, all models achieved a comparable level of reward and models convergence. The only distinguishing factors were variations in training duration and model dimensions. As our model employs two-layer MLPs, differences in computational complexity based on the number of hidden layer nodes can also be referred. Although the original MAUP with 128 nodes is faster than the tiny MAUP with 16 nodes, but the energy consumption of the tiny MAUP is much less than that of the original MAUP in terms of total training energy consumption. Because the time complexity analysis shows that the time complexity of the original MAUP is much larger than that of the tiny MAUP. And considering future deployment of the model, this leads to significant improvements in reducing static random-access memory (SRAM) space occupation.

The convergence of the MAUP algorithm under different learning rates is illustrated in Fig. 3. Specifically, learning rates of 0.0001, 0.0002, 0.0005, and 0.001 were employed for comparison purposes [Fig. 3(a)–(d)]. As shown in Fig. 3(a), when a small learning rate of lr = 0.0001 is used, the algorithm oscillates initially, but gradually converges around 18 000 epochs. Fig. 3(b) shows that when the learning rate is 0.0002, the algorithm also oscillates initially, but the reward keeps improving with some fluctuations in the middle, and finally stabilizes around 20 000 epochs. When the learning rate is lr = 0.0005, the overall convergence is basically the same as that of lr = 0.0002. Compared with the learning rate of 0.0001, using these two learning rates can also converge to the same results, but the convergence speed is slower than that of lr = 0.0001. When the learning rate is 0.001, due to the exploration process at the beginning, the algorithm oscillates significantly at first. At around 10 000 epochs, the reward improves, and the algorithm shows signs of convergence at around 15 000 epochs. The reward then improves again, and the result converges at around 18 000 epochs. Therefore, using a larger learning rate can also achieve convergence, but the initial oscillation is more significant. A smaller learning rate can speed up the convergence speed. Therefore, the learning rate in the final simulation experiment was set to 0.0001.

The convergence of the algorithm under different soft update coefficients is illustrated in Fig. 4, with the soft update coefficients set to 0.01 and 0.1, respectively. As depicted in the figure, when the soft update coefficient is set to 0.01, the algorithm experiences initial instability due to the learning process. However, it gradually converges after approximately 18 000 training steps. On the other hand, when using a soft update coefficient of 0.1, persistent instability occurs during early stages of training before eventually converging at around 32 000 training steps; albeit at a relatively slower rate compared to smaller coefficients. These results indicate that larger soft update coefficients have a more pronounced impact on experimental outcomes, while demonstrating faster convergence for MAUP algorithm under smaller values of this coefficient. Consequently, a final experiment was conducted with a soft renewal coefficient set as 0.01.

As shown in Fig. 5, we investigated the variation of the UAV's continuous operation time during the training process. This study assumes that once the UAV leaves the mission area, the training of the current round stops, and a certain penalty is imposed. The UAV needs to learn to adopt a reasonable flight strategy to stay within the designated area in order to obtain a higher reward. From the figure, it can be seen that in the early stage of training, the UAV occasionally goes out of bounds, causing the round to terminate before completing two

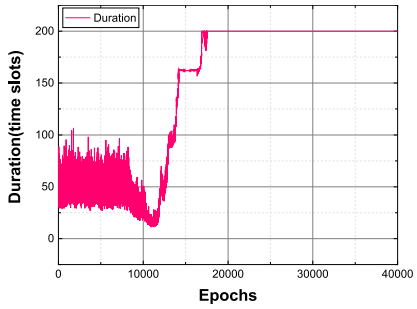


Fig. 5. Variation of duration with the training process.

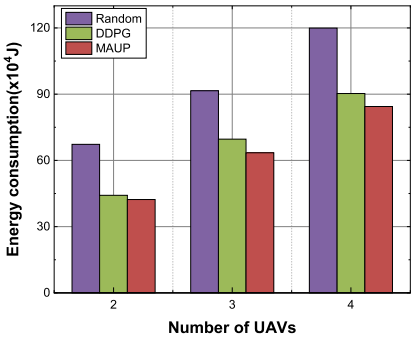


Fig. 6. Energy consumption under different number of UAVs.

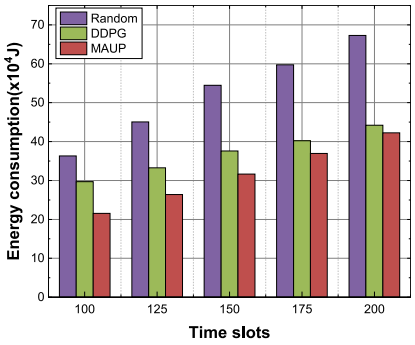


Fig. 7. Impact of the number of time slots on energy consumption.

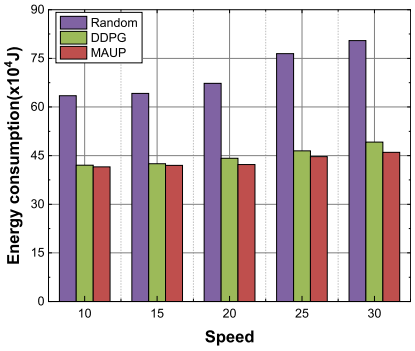


Fig. 8. Impact of the UAV flight speed on energy consumption.

hundred time slots, resulting in large fluctuations in continuous operation time in the early stage. When the number of epochs is around 18 000, the UAV can fly for a full 200 time slots without going out of bounds. After that, there will be no situation where the UAV flies out of the designated area. This also means that the UAV has learned how to fly reasonably and stay within the designated boundary area in the later stage.

The experimental results for varying numbers of UAVs are investigated in Fig. 6. As illustrated in Fig. 6, the random method exhibits higher energy consumption compared to the other two methods. With an increasing number of drones, all methods experience an increase in energy consumption. However, MAUP still demonstrates relatively low-energy consumption relative to the other two methods, while the random method consistently yields the highest energy consumption. This can be attributed to the fact that actions in the random method are selected randomly and do not represent a comparatively superior strategy. Nevertheless, as drone count increases, there is a gradual deterioration in effectiveness observed with DDPG method; indicating that drone count has a certain influence on DDPG's performance and suggesting that our proposed approach is more suitable for this scenario.

The impact of mission duration (number of time slots) on the final outcomes for a UAV count of two is illustrated in Fig. 7. The number of time slots ranges from 100 to 200. As depicted in Fig. 7, the total energy consumption of the UAV escalates with an increase in the number of time slots, indicating a prolonged execution period for the traffic mission. Nevertheless, our proposed method surpasses both DDPG and random methods.

The energy consumption of the UAV at different flight speeds is illustrated in Fig. 8. This study investigates the energy consumption of the UAV within a range of flight speeds from 10 to 30 m/s. It can be observed that at lower speeds, specifically 10 and 15 m/s, the overall energy consumption remains relatively low with minimal variation. However, as the UAV's speed increases, there is a corresponding increase in its overall energy consumption.

VI. Conclusion

In this article, the path planning problem of UAVs performing road traffic monitoring tasks with limited range is studied for the main purpose of optimizing the flight and hovering energy consumption of UAVs. By analyzing the energy consumption problem during UAV traffic monitoring, the energy optimization problem is defined. Subsequently, through an analysis of the optimization problem, the UAV energy optimization problem is transformed into a reinforcement learning problem, and a UAV path planning algorithm based on MADDPG is designed. Subsequently, we proceed to adjust the neurons in the hidden layers and perform pruning based on L1 norm pruning. Finally, simulation experiments are conducted to compare the proposed algorithm with other algorithms in order to validate its effectiveness. The size of our model is smaller compared to the standard one. The simulation results demonstrate that the algorithm presented in this article can effectively reduce the energy consumption of UAVs during flight and hovering. Experimental results also point to the suitability of the MADDPG method for designing scenarios with a set of sequential decision problems.

However, this study only considers the energy consumption of UAVs during flight and hovering, without taking into

account the communication energy consumption of the UAVs. Additionally, this article does not consider the issue of task offloading for UAVs, and the optimization objective only focuses on energy consumption, implying a single-objective optimization problem. In future research, it would be beneficial to consider task offloading, taking into account both the path planning problem for UAVs during traffic monitoring and the offloading of information collected by the UAVs to nearby edge servers. Moreover, a multiobjective optimization problem could be designed, incorporating not only the optimization of UAV energy consumption but also the consideration of Age of Information (AoI). For the Tiny ML component, we will first train a large-scale model, followed by conducting model compression and subsequently comparing its effect with that of a smaller model.

REFERENCES

[1] T. Szydlo, P. P. Jayaraman, Y. Li, G. Morgan, and R. Ranjan, "TinyRL: Towards reinforcement learning on tiny embedded devices," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manag.*, 2022, pp. 4985–4988.

[2] M. Shafique, T. Theocharides, V. J. Reddy, and B. Murmann, "TinyML: Current progress, research challenges, and future roadmap," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, 2021, pp. 1303–1306.

[3] X. Kong, Y. Wu, H. Wang, and F. Xia, "Edge computing for Internet of Everything: A survey," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23472–23485, Dec. 2022.

[4] A. K. Bandani, S. Riyazuddien, P. Bidare Divakarachari, S. N. Patil, and G. Arvind Kumar, "Multiplicative long short-term memory-based software-defined networking for handover management in 5G network," *Signal, Image Video Process.*, vol. 17, no. 6, pp. 2933–2941, 2023.

[5] A. Alioua, H.-E. Djeghri, M. E. T. Cherif, S.-M. Senouci, and H. Sedjelmaci, "UAVs for traffic monitoring: A sequential game-based computation offloading/sharing approach," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107273.

[6] H. Liu et al., "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 4125–4129.

[7] B. Parameshachari, S. Gurumoorthy, J. Frnda, S. C. Nelson, and K. R. Balmuri, "Cognitive linear discriminant regression computing technique for HTTP video services in SDN networks," *Soft Comput.*, vol. 26, no. 2, pp. 621–633, 2022.

[8] R. Jiang et al., "Deepurbanevent: A system for predicting citywide crowd dynamics at big events," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2019, pp. 2114–2122.

[9] L. P. Qian, H. Zhang, Q. Wang, Y. Wu, and B. Lin, "Joint multi-domain resource allocation and trajectory optimization in UAV-assisted maritime IoT networks," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 539–552, Jan. 2023.

[10] X. Zhou et al., "Edge-enabled two-stage scheduling based on deep reinforcement learning for Internet of Everything," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295–3304, Feb. 2022.

[11] X. Kong et al., "Deep reinforcement learning-based energy-efficient edge computing for Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6308–6316, Sep. 2022.

[12] 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.

[13] X. Kong, W. Zhang, Y. Qu, X. Yao, and G. Shen, "FedAWR: An interactive federated active learning framework for air writing recognition," *IEEE Trans. Mobile Comput.*, early access, Sep. 28, 2023, doi: 10.1109/TMC.2023.3320147.

[14] N. A. Khan, N. Jhanjhi, S. N. Brohi, R. S. A. Usmani, and A. Nayyar, "Smart traffic monitoring system using unmanned aerial vehicles (UAVs)," *Comput. Commun.*, vol. 157, pp. 434–443, May 2020.

[15] T. Lyu, H. Zhang, and H. Xu, "Resource allocation in UAV-assisted wireless powered communication networks for urban monitoring," *Wireless Commun. Mobile Comput.*, vol. 2022, Aug. 2022, Art. no. 7730456.

[16] N. Bashir and S. Boudjit, "An energy-efficient collaborative scheme for UAVs and VANETs for dissemination of real-time surveillance data on highways," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2020, pp. 1–6.

[17] N. Bashir, S. Boudjit, and S. Zeadally, "A closed-loop control architecture of UAV and WSN for traffic surveillance on highways," *Comput. Commun.*, vol. 190, pp. 78–86, Jun. 2022.

[18] H. Huang, Y. Yang, H. Wang, Z. Ding, H. Sari, and F. Adachi, "Deep reinforcement learning for UAV navigation through massive MIMO technique," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1117–1121, Jan. 2020.

[19] W. Zhang, Q. Wang, X. Liu, Y. Liu, and Y. Chen, "Three-dimension trajectory design for multi-UAV wireless network with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 600–612, Jan. 2020.

[20] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May 2020.

[21] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "UAV path planning for wireless data harvesting: A deep reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. GLOBECOM*, 2020, pp. 1–6.

[22] Z. Ning, Y. Yang, X. Wang, Q. Song, L. Guo, and A. Jamalipour, "Multi-agent deep reinforcement learning based UAV trajectory optimization for differentiated services," *IEEE Trans. Mobile Comput.*, early access, Sep. 5, 2023, doi: 10.1109/TMC.2023.3312276.

[23] X. Wang, Z. Ning, S. Guo, M. Wen, L. Guo, and V. Poor, "Dynamic UAV deployment for differentiated services: A multi-agent imitation learning based approach," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2131–2146, Apr. 2023.

[24] N. O. Lambert, D. S. Drew, J. Yaconelli, S. Levine, R. Calandra, and K. S. Pister, "Low-level control of a quadrotor with deep model-based reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4224–4230, Oct. 2019.

[25] A. Molchanov, T. Chen, W. Hönig, J. A. Preiss, N. Ayanian, and G. S. Sukhatme, "Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 59–66.

[26] B. P. Duisterhof et al., "Tiny robot learning (tinyRL) for source seeking on a nano quadcopter," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 7242–7248.

[27] T. M. Ho, K.-K. Nguyen, and M. Cheriet, "UAV control for wireless service provisioning in critical demand areas: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 7138–7152, Jul. 2021.

[28] K. Kang, S. Belkhale, G. Kahn, P. Abbeel, and S. Levine, "Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6008–6014.

[29] R. Liu, M. Xie, A. Liu, and H. Song, "Joint optimization risk factor and energy consumption in IoT networks with TinyML-enabled Internet of UAVs," *IEEE Internet Things J.*, early access, Jan. 1, 2023, doi: 10.1109/JIOT.2023.3348837.

[30] G. Chopra, S. Rani, W. Viriyasitavat, G. Dhiman, A. Kaur, and S. Vimal, "UAV-assisted partial co-operative NOMA based resource allocation in C2VX and TinyML based use case scenario," *IEEE Internet Things J.*, early access, Jan. 9, 2023, doi: 10.1109/JIOT.2024.3351733.

[31] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[32] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1226–1252, 2nd Quart., 2021.

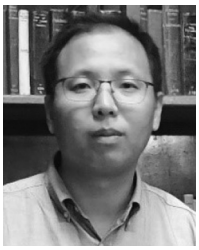
[33] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Appl. Intell.*, vol. 53, pp. 13677–13722, Jun. 2023.

[34] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[35] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[36] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 73–84, Mar. 2021.

[37] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 157–163.



Xiangjie Kong (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively.

He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. Previously, he was an Associate Professor with the School of Software, Dalian University of Technology, Dalian, China. He has published over 200 scientific papers in international journals and conferences (with over 170 indexed by ISI SCIE). His research interests include network science, knowledge discovery, and urban computing.

Prof. Kong is a Distinguished Member of CCF and a member of ACM.

Chenhao Ni received the M.Sc. degree from Tianjin University of Technology, Tianjin, China, in 2023. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China.

His research interests include mobile computing, Internet of Vehicle, deep reinforcement learning, and tiny ML.

Gaohui Duan received the B.Sc. degree in software engineering from Henan University of Science and Technology, Luoyang, China, in 2020. He is currently pursuing the master's degree with the School of Software, Dalian University of Technology, Liaoning, China.

His research interests include urban data mining, network embedding, and data set completion.

Guojiang Shen received the B.Sc. degree in control theory and control engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively.

He is currently a Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. His current research interests include artificial intelligence theory, big data analytic, and intelligent transportation systems.

Yao Yang received the bachelor's and Ph.D. degrees from the Department of Physics, Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2011, respectively.

He is currently a Research Expert with Zhejiang Laboratory, Hangzhou, China. He has published more than 20 research papers in top internationals and conferences. His research interests include deep representation learning, federated learning, causal inference, and interdisciplinary topics of machine learning and physics.

Sajal K. Das (Fellow, IEEE) received B.S. degree in computer science and engineering from Calcutta University, Kolkata, India, the M.S. degree in computer science and automation from the Indian Institute of Science, Bengaluru, India, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, FL, USA.

He is a Curators' Distinguished Professor of Computer Science and the Daniel St. Clair Endowed Chair with Missouri S&T, Rolla, MO, USA. His research interests include cyber–physical systems, Internet of Things, machine/federated learning, smart environments, wireless sensor networks, pervasive and mobile computing, and cybersecurity.

Dr. Das is the Editor-in-Chief of *Pervasive and Mobile Computing* journal (Elsevier) and an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and IEEE/ACM TRANSACTIONS ON NETWORKING.