Live Traffic Video Multicasting Services in UAV-Assisted Intelligent Transport Systems: A Multiactor Attention Critic Approach

Fang Fu, Bin Xue, Lin Cai, *Fellow, IEEE*, Laurence T. Yang, *Fellow, IEEE*, Zhicai Zhang, *Member, IEEE*, Jia Luo, and Chenmeng Wang

***Abstract*—Live traffic video is vitally important for vehicles in future intelligent transport systems (ITSs). Due to the limitation of onboard sensors, vehicles may not be able to obtain a full view of the traffic situations which endangers safety for autonomous driving vehicles. In this article, we propose a traffic video multicasting scheme by using video splitting and group splitting techniques for unmanned aerial vehicles (UAVs)-assisted ITS, in which UAVs are considered as the eyes in the sky to capture real-time traffic videos. We aim to maximize the long-term video quality received by vehicles by jointly optimizing vehicle grouping and spectrum allocation. Considering the interactions among UAVs, the above optimization problem is formulated as a multiagent coordination problem in the form of a Markov game (MG). The MG is subsequently solved by leveraging a state-of-the-art multiagent deep reinforcement learning (MADRL) algorithm, namely, multiactor attention critic (MAAC), in which an attention mechanism is utilized to pay attention to other agents to make the learning process more effective and scalable. Extensive simulation results show that the MAAC-based algorithm has better performance in terms of video quality and spectrum efficiency compared with the baseline methods.**

***Index Terms*—Deep reinforcement learning (DRL), evolved multimedia broadcast multicast service (eMBMS), multiagent learning, traffic videos.**

Manuscript received 5 February 2023; revised 7 May 2023; accepted 24 May 2023. Date of publication 5 June 2023; date of current version 7 November 2023. This work was supported by the National Key Research and Development Program of China under Grant 2022ZD0118300. *(Corresponding author: Zhicai Zhang.)*

Fang Fu, Zhicai Zhang, and Chenmeng Wang are with the School of Computer Science and Technology, Hainan University, Haikou 570228, China (e-mail: 962005326@qq.com; 215780778@qq.com; wangc@hainanu.edu.cn).

Bin Xue is with the School of Physics and Electronic Engineering, Shanxi University, Taiyuan 030006, Shanxi, China (e-mail: 815020702@qq.com).

Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: cai@ece.uvic.ca).

Laurence T. Yang is with the School of Computer Science and Technology, Hainan University, Haikou 570228, China, also with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: ltyang@ieee.org).

Jia Luo is with the School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: luojia@cqupt.edu.cn).

Digital Object Identifier 10.1109/JIOT.2023.3282936

I. Introduction

REAL time, accurate, and comprehensive traffic information is essential for driving safety of connected autonomous vehicles (CAVs) [1]. As a key component of the future intelligent transport systems (ITSs), CAVs can both sense and collect traffic data. Each CAV's own onboard sensors, such as cameras, LiDAR, radar, GPS, etc. can gather driving data and traffic information [2], [3]. Advanced communication technologies, such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) communications [4], are utilized to collect data from the surrounding environment. However, due to the limitation of locations, numbers, and qualities of the sensing devices, the traffic situations may not be fully observed by vehicles, leading to safety concerns of CAVs.

Unmanned aerial vehicles (UAVs) assisted ITS, a new paradigm of the smart city is emerged recently [5], [6], which is a promising solution for numerous traffic video applications, such as automatic driving, traffic monitoring and management, and traffic accident rescue. In automatic driving scenarios, the global perspective of traffic images captured by high definition (HD) camera mounted on UAVs can be relayed to CAVs to assist them in traffic guidance, traffic activity analysis, identify objects, and help cruise control systems to make sensible decisions. Nevertheless, live HD video data are costly to share due to high resolution, high frame rate, and highly dynamic range characteristics. Furthermore, since the same traffic video content is delivered by the UAV to all CAVs subscribed simultaneously, traditional unicast transmission mode is spectrum inefficiency, especially in heavy traffic areas. Hence, resource allocation is challenging.

Evolved multimedia broadcast multicast service (eMBMS) is introduced by third generation partnership program (3GPP) [7], to serve more users for data/video applications with limited radio resources using conservative multicast scheme (CMS) [8] or opportunistic multicast scheme (OMS) [9]. In CMS, a live content is multicasted/broadcasted to all clients interested through a shared channel. However, due to the high heterogeneity of links, especially in high dynamic ITS networks, the transmission rate is bounded by the user with the worst channel conditions in a multicasting group, CMS suffers low spectrum efficiency consequently. Vice versa, OMS aims to leverage the multiuser diversity by selecting only a part of multicast users to maximize throughput at each scheduling. However, the short-term fairness among users cannot be guaranteed, which cannot be used in traffic video multicasting scenarios.

In this article, we take advantage of group splitting and stream splitting techniques for traffic video multicasting services in UAVs-assisted ITS. First, by employing advanced stream splitting methods, such as scalable video coding (SVC) encoder, the source video captured by the UAV is split into multiple substreams with lower data rate, including a base layer (BL), and a couple of enhancement layers (ELs) that can be transmitted independently by dedicated channels. Second, by leveraging group splitting methods, vehicles are split into different groups according to channel conditions. To be specific, considering fairness and efficiency, all vehicles should receive the base substream while vehicles with good channel conditions are split into different groups to receive full/a part of enhancement substreams to improve video quality. Third, a vehicle with good channel conditions may belong to multiple groups and receives different layers' data independently that will be combined together later. In such a manner, the "basic" quality content is multicasted to all vehicles and the "improved" quality content is only transmitted to those with better channel conditions [10]. In this novel structure, we have to answer the following questions. 1) How to jointly optimize user grouping and spectrum allocation?; 2) How to guarantee that each layer can be decoded successfully? Because each layer data is multicasted independently and the *i*th layer data can be decoded successfully if and only if the data of all lower layers have been correctly received.

To address these open issues associated with stream splitting and group splitting, we propose a multiagent deep reinforcement learning (MADRL)-based traffic video multicasting scheme for UAVs-assisted ITS. The main contributions of this article are listed as follows.

1) We consider multiple UAVs traffic video multicasting scenarios, where the interference caused by other UAVs is considered. Most of the existing works only considered single-cell scenarios for eMBMS applications, are not directly applicable to multiple UAVs scenarios.
2) To maximize the long-term video quality, we jointly optimize user grouping and physical resource blocks (PRBs) assignment of the whole networks. The weighted sum of all layers' validity functions is defined as the optimization objective, where the weights are utilized to measure the importance of different layers and the validity function is introduced to measure whether a layer can be decoded successfully or not.
3) We leverage a novel MADRL method, namely, multiactor attention critic (MAAC), to deal with the problem. The considered problem is a multiagent policy optimization problem, where each agent needs to cooperate with other agents and makes a sequence of actions to maximize the long-term reward. Since the traditional methods, such as static optimization theory, aim to search the optimal/suboptimal solution to maximize the short-term reward according to the current observation, they are not feasible. Besides, to avoid curse of dimensionality caused by traditional DRL methods when the number of agents is large, we adopt MAAC to solve the problem. In MAAC, an attention mechanism is utilized to pay attention to other agents, which enables the agents to coordinate with each other effectively and makes the learning process more scalable.
4) A Python-based simulator is developed to implement the proposed algorithm and other baseline methods. Extensive simulation results show the effectiveness of the proposed scheme.

The remainder of this article is organized as follows. The related work is discussed in Section II. The system model and assumptions are presented in Section III. We formulate the optimization problem in Section IV, which is subsequently reformulated as a Markov game (MG). In Section V, we adopt MAAC to solve the MG. We present the simulation results and related discussions detailedly in Section VI. Finally, Section VII gives the conclusion and future works.

II. Related Work

We discuss the related work of video streaming services in eMBMS systems, UAVs-assisted ITS, and DRL-based resource optimization, respectively.

A. Video Streaming Services in eMBMS Systems

Live stream multicasting is the main application in eMBMS systems and many excellent works have been done in literature. For example, Montalban et al. [11] proposed a layer-division multiplexing (LDM)-based subgrouping method for multimedia multicasting services in cellular networks. Araniti et al. [12] presented an original multicast resource allocation method based on subgroup formation and application layer joint coding to improve the performance of live streaming in satellite-eMBMS systems in terms of throughput and video Quality of Experience (QoE). Zhang et al. [13] proposed a novel multicast framework by taking advantage of dynamic adaptive streaming over HTTP (DASH) and cloud radio access networks (CRANs), and then, user grouping, bitrate selection, and spectrum allocation were jointly optimized to enhance the QoE of multicasting services. However, the spectrum efficiency of these works [11], [12], [13] can be further improved by leveraging the advanced stream splitting techniques.

For multicast live stream applications, group splitting could draw support from stream splitting techniques. For example, Condoluci et al. [14] proposed a novel group splitting-based spectrum resource allocation scheme for SVC stream multicasting services in a single-cell scenario. A QoE driven power allocation algorithm was proposed for scalable video multicast services in a nonorthogonal multiple access (NOMA)-based single-cell scenario [15]. Zhu et al. [16] proposed a NOMA enhanced SVC multicast method to improve the video QoE in cellular networks. The results of [14], [15], and [16] show that by jointly leveraging group splitting and stream splitting techniques, eMBMS systems have better performance in terms of fairness and efficiency in video multicasting applications.

Nevertheless, most of the existing works [11], [12], [13], [14], [15], [16] considered single-cell scenarios, where intercell interference caused by neighbor cells was ignored. In addition, all aforementioned works are dealing with resource optimization problem for live stream multicasting services in traditional wireless networks, which cannot be directly utilized in UAVs-assisted ITS, since both the wireless channel and network topology are highly dynamic and more complicated.

B. UAVs-Assisted ITS

Menouar et al. [5] first introduced the concept of UAVs-assisted ITS in 2017, which hereupon induced widely discussion in industry and academia. These existing works are developed from the following perspectives.

First, thanks to the flexibility and mobility, UAVs can serve as temporary base stations/edge servers, to provide emergence communications [17], [18], edge computing [19], and caching services for vehicles [20], in areas where terrestrial infrastructure is temporarily unavailable or insufficient, such as highways [17], hot spots [18], disaster areas [21], etc. For example, Samir et al. [17] presented a DRL based UAVs' trajectory planning scheme aiming to provide full coverage for vehicles on highways. Oubbati et al. [18] proposed a UAV-assisted routing strategy to improve the connectivity of vehicular ad hoc networks in urban areas. Han et al. [22] introduced a novel bilevel optimization method to jointly optimize UAV deployment and path planning for UAV-enabled networks. Besides, since autonomous vehicles require a huge amount of computing and caching resources to process traffic data and make decisions, UAVs can act as flying mobile-edge computing (MEC) servers to provide computing and caching services for vehicles. For example, Zhao et al. [19] proposed a UAV-enabled vehicular computation offloading scheme, where UAVs acted as MEC servers to execute computing intensive and time sensitive tasks for ground vehicles. Al-Hilo et al. [20] presented a UAV caching system, where the UAV served as an MEC server to help vehicles downloading/storing the target contents.

Second, UAVs can carry sensors to collect and process real-time traffic information for vehicles. For example, Jian et al. [23] proposed a UAV enabled traffic data collection and congestion recognition system. To guarantee the freshness of traffic data gathered by UAVs, an Age of Information (AoI) aware trajectory optimization algorithm was proposed for UAVs enabled ITS [24]. A lightweight blockchain-based data sharing scheme was proposed to guarantee the security of data perceived by UAVs and vehicles in disaster area [21]. Khan et al. [25] presented a framework for processing and analyzing the traffic flow at urban roundabouts based on the images captured by the UAV.

However, when UAV provide live traffic multicasting services for vehicles, how to jointly optimize user grouping and resource allocation problems in multi-UAV scenarios remains an open issue, which motivates this article.

C. DRL-Based Resource Optimization

Deep reinforcement learning (DRL) methods are widely used in resource allocation and UAV path planning fields [26], [27], [28]. For instance, Du et al. [26] presented an asynchronous advantage actor–critic (A3C) based communication and computing resources allocation algorithm to maximize miners' profits in MEC-assisted blockchain systems. Fu et al. [27] introduced a soft actor–critic based resource optimization scheme for live video transcoding and transmitting tasks in fog computing enabled vehicular networks. Zhang et al. [28] proposed an energy-efficient video transmission scheme for UAV-assisted wireless networks, where a safe deep Q-learning network (safe-DQN) was leveraged to optimize the UAV's trajectory, video version selection, and power control. Since all agents are collectively treated as a single agent in these methods [26], [27], [28], which results in the action space increasing exponentially w.r.t. the number of agents, the scalability is poor [29].

To overcome the above weakness of the single-agent DRL methods, MADRL is widely used in wireless resource optimization recently. Dai et al. [30] proposed a MADRL-based UAV deployment and resource allocation scheme for UAV networks, where each UAV was considered as an independent agent to optimize the policy individually based on partial observations. A joint subchannel allocation and power control algorithm based on MADRL was proposed to maximize the total throughput of V2I links in vehicular networks [31], in which each agent made decision based on local channel state information. Zhang et al. [32] introduced a multiagent Q-learning algorithm based on Stackelberg game framework to optimize power control of wireless networks. The reinforcement learning methods used in [30], [31], and [32] are simple ways to execute multiagent tasks. In these approaches, each agent makes decision independently based on its local observations, where the actions of other agents are treated as part of environment. Nevertheless, considering the changing policies of opponents, the environment of each agent may be dynamic and non-Markovian, which violates the basic assumption of DRL and makes the learning process inefficient [33], [34], [35], [36].

MAAC combines the strong points of single-agent DRL [26], [27], [28] and independent reinforcement learning [30], [31], [32], where a critic evaluates state/state-action values in a global view based on the information from all agents and the actors, however, train the policy networks and take actions solely according to their own local observations. Since each agent does not need other agents' actions information in MAAC when training and executing the policies, the challenges of nonstationary and non-Markovian can be circumvented. As an improvement of multiagent deep deterministic policy gradient (MADDPG) [37], MAAC has the following advantages.

1) A multiagent DRL framework is employed, in which actors (UAVs) are trained distributedly based on centrally computed advantage function by a shared critic [the roadside units (RSUs)] and the critic is trained centrally to avoid the nonstationary and non-Markovian environment states caused by other agents' time-varying

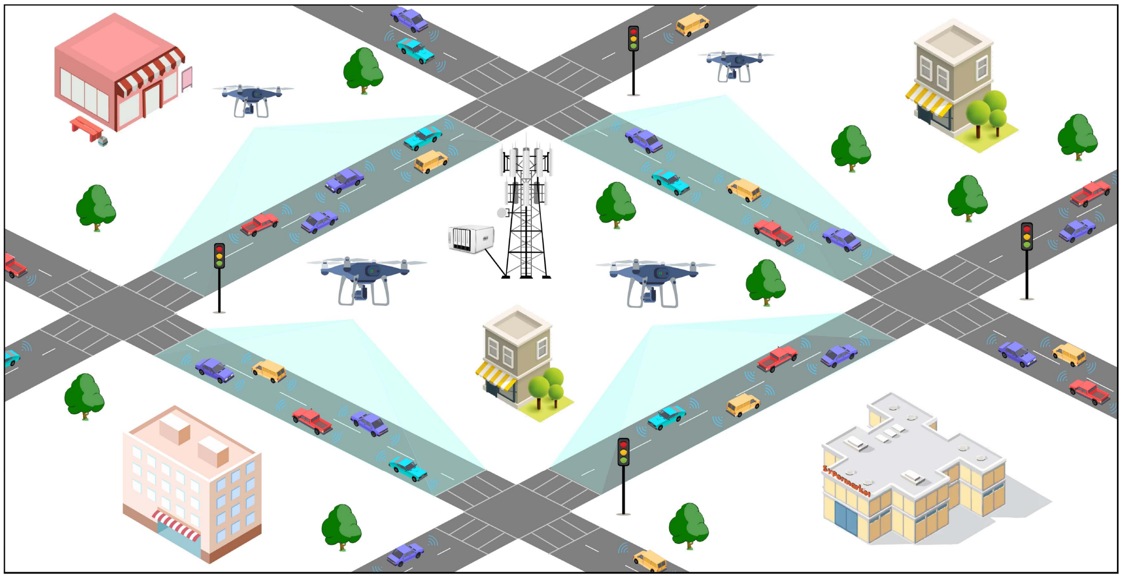


Fig. 1. UAV-assisted ITS scenario.

policies. And then, each actor executes its own policy based on the local observation that reduces the communication overhead with other agents.

2) An attention mechanism is utilized to select relevant information for each agent. Paying attention to specific agents, such as their neighbors, makes the learning process more effective and scalable [38], [39].
3) Soft actor–critic is leveraged by each agent, which aims to maximize the long-term entropy objective to ensure stability and exploration.
4) A counterfactual baseline function is employed to calculate the advantage function to reduce the variance and solve the credit assignment problem in multiagent settings.

Above all, we will employ MAAC to solve the user grouping and PRBs assignment problem in this article.

III. System Model and Assumptions

Fig. 1 is the UAV-assisted ITS scenario considered in this article. Multiple UAVs equipped with HD cameras and communication modules are hovering above the road to provide traffic information for vehicles. The road is split into several segments and each UAV is responsible for providing real-time traffic video multicasting services for the vehicles located on a road segment. Time is slotted equally and the duration of one time slot is Δt s. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the set of time slots. We denote $\mathcal{N} = \{1, \ldots, N\}$ as the set of UAVs and the coordinate of UAV n is $[X_n, Y_n, Z_n]$ $(\forall n \in \mathcal{N})$. The set $\mathcal{V}_n = \{v | v = 1, \ldots, V_n\}$ is vehicles on the road segment served by UAV n and $[X_{n,v}(t), Y_{n,v}(t), Z_{n,v}(t)]$ $(\forall v \in \mathcal{V}_n)$ is the coordinates of vehicle v at the tth time slot. RSUs are deployed along the road that can provide computing services for UAVs.

A. SVC Streaming Model

The traffic video captured by the UAV is encoded by a SVC encoder, where the stream is composed of a series of video sequences, and the duration of each sequence is $\iota = \Delta t$ s. In SVC standard, the video sequence is encoded into a group of picture (GOP), each GOP contains I numbers of layers: a BL and $(I - 1)$ ELs. The BL stream is able to decode itself and several ELs data can be appended to the BL stream to enhance the video quality, i.e., videos with more ELs means higher video quality. Correspondingly, the video qualities of SVC streams are classed into I levels. Vehicles with harsh channel conditions can only receive the BL data of traffic videos and vehicles with good channel conditions can receive a BL and all/a portion of ELs.

B. User Grouping Model

To save the spectrum resource, we employ the eMBMS technology to multicast streams. Considering that the video qualities have I levels, vehicles will be clustered into I groups, vehicles in each group can receive the same layer stream data. To be specific, let $\mathcal{G}_n = \{G_{n,i} | i \in \mathcal{I}\}$ denote the set of groups served by UAV n, where $\mathcal{I} = \{1, \ldots, I\}$ and vehicles in group $G_{n,i}$ can only receive the ith layer data multicasted by the UAV. Let $\boldsymbol{C}_n^t = \{c_{n,i}^v | \forall i \in \mathcal{I}, v \in \mathcal{V}_n\}$ denote vehicle association strategy of UAV n, the element $c_{n,i}^v = 1$ represents that vehicle v is in group $G_{n,i}$, and $c_{n,i}^v = 0$ otherwise. Since a vehicle may receive multilayers data, the vehicle will be in multiple groups correspondingly. Fig. 2 is an example of vehicle grouping diagram, where vehicle v_1 with high channel quality is associated with G_1, G_2, and G_3 and can receive BL, EL1, and EL2 data at the same time slot, respectively, and these different layers are combined together to produce the final stream by vehicles [40]. On the other hand, vehicle v_5 with a harsh channel condition is classified uniquely with G_1 and receives the BL data. Considering fairness and efficiency, we try to guarantee that each vehicle can receive the BL data, i.e., all vehicles are in G_1 as shown in Fig. 2, while the vehicles with good channel conditions may also be in other groups to acquire higher layer

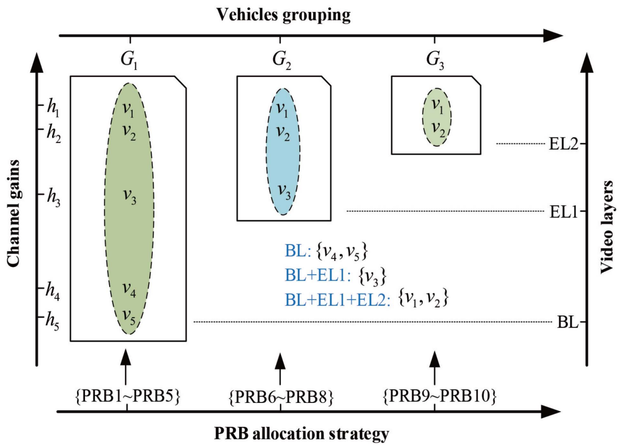


Fig. 2. Vehicle grouping diagram.

data. Therefore, $c_{n,i}^{v}$ should satisfy the following condition:

$$1 \leq \sum_{i=1}^{I} c_{n,i}^{v} \leq I \; \forall c_{n,i}^{v} \in \{0,1\}, v \in \mathcal{V}_n, n \in \mathcal{N}. \tag{1}$$

C. UAV to Vehicles Channel Model

The channel from UAV n to vehicle v is denoted by $g_{n,v}$ that is calculated by [22]

$$g_{n,v} = 20\log\left(4\pi f_c d_{n2v} c^{-1}\right) + \xi_{\text{LoS}} P_{\text{LoS}} + \xi_{\text{NLoS}}(1 - P_{\text{LoS}}) \tag{2}$$

where f_c is the carrier frequency, c is the speed of light, and d_{n2v} is the distance from UAV n to vehicle v and calculated by $d_{n2v} = \sqrt{(X_n - X_{n,v}(t))^2 + (Y_n - Y_{n,v}(t))^2 + (Z_n)^2}$. ξ_{LoS} and ξ_{NLoS} are the additional mean losses of Line-of-Sight (LoS) and non-LoS (NLoS) links, respectively [41]. The probability of LoS is calculated by a Sigmoid function, i.e., $P_{\text{LoS}}(\beta) = [1 + \delta \exp(-\mu(\beta - \delta))]^{-1}$, where δ and μ are S-curve parameters [42], and β is given by $\beta = (180/\pi)\arctan(Z_n^{-1}\sqrt{(X_n - X_{n,v}(t))^2 + (Y_n - Y_{n,v}(t))^2})$.

D. Multicasting Transmission Model

As we have stated previously, UAVs act as aerial base stations to multicast the traffic videos to vehicles to enhance their automatic driving abilities. All vehicles in a multicast group are treated as a single entity by the UAV and served on the same channels. To avoid intracell interference, we adopt orthogonal frequency division multiple access (OFDMA) technology. Let $\mathcal{K} = \{1, \ldots, K\}$ represent the set of available PRBs of the system. We employ $\boldsymbol{H}_n^t$ ($n \in \mathcal{N}$) to denote the PRB allocation strategy of UAV n, which is an $I \times K$ matrix with binary variables $\eta_{n,i}^{k}$ ($\forall i \in \mathcal{I}$, $k \in \mathcal{K}$). Since each PRB can only be utilized by at most one group associated with UAV n during each time slot, the variable $\eta_{n,i}^{k}$ satisfies

$$\eta_{n,i}^{k} \in \{0,1\} \; \forall i \in \mathcal{I}, k \in \mathcal{K}, n \in \mathcal{N} \tag{3}$$

$$\sum_{i=1}^{I} \eta_{n,i}^{k} \leq 1 \; \forall k \in \mathcal{K}, n \in \mathcal{N} \tag{4}$$

where $\eta_{n,i}^{k} = 1$ denotes that PRB k is occupied by group $G_{n,i}$ and $\eta_{n,i}^{k} = 0$ conversely.

Let $\boldsymbol{\mathcal{H}}_{-n}^{t} = \{\boldsymbol{H}_m^t | \forall m \in \mathcal{N}, m \neq n\}$ denote the set of PRB allocation strategies of all UAVs with the exception of UAV n, where $\boldsymbol{H}_m^t$ is the strategy of UAV m. The interference suffered by vehicle v of UAV n on PRB k is denoted as

$$\Gamma_{n,v}^{k}\left(\boldsymbol{\mathcal{H}}_{-n}^{t}\right) = \sum_{m=1, m\neq n}^{N} \eta_m^k \cdot p_m^k g_{m,v} \tag{5}$$

where $\eta_m^k = \sum_{i \in \mathcal{I}} \eta_{m,i}^{k}$ represents whether PRB k is occupied by UAV m or not, i.e., $\eta_m^k = 1$ denotes that PRB k is occupied by UAV m and $\eta_m^k = 0$ otherwise. p_m^k denotes the transmit power of UAV m on PRB k and $g_{m,v}$ is the channel gain between UAV m and vehicle v ($\forall v \in \mathcal{V}_n$).

The maximal downloading rate achieved by vehicle v on PRB k is calculated by

$$\mathcal{R}_{n,v}^{k}\left(\boldsymbol{H}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right) = \eta_{n,i}^{k} \cdot \omega \log_2\left(1 + \frac{p_n^k g_{n,v}}{\sigma^2 + \Gamma_{n,v}^{k}\left(\boldsymbol{\mathcal{H}}_{-n}^{t}\right)}\right) \tag{6}$$

which is a function of $\boldsymbol{H}_n^t$ and $\boldsymbol{\mathcal{H}}_{-n}^{t}$. In (6), ω is the frequency bandwidth of each PRB, p_n^k and $g_{n,v}$ are the transmit power of UAV n on PRB k and channel gain between UAV n and vehicle v, respectively. σ^2 is the additive white Gaussian noise power. Since the transmit rate of multicasting services depends on the worst link [13], the data rate of vehicles associated with group $G_{n,i}$ on PRB k is calculated by

$$\mathcal{R}_{n,i}^{k}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right) = \min_{v \in \mathcal{V}_n}\left\{c_{n,i}^{v} \cdot \mathcal{R}_{n,v}^{k}\left(\boldsymbol{H}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right)\right\}. \tag{7}$$

Therefore, the total transmit rate of vehicles in group $G_{n,i}$ is calculated by

$$\mathcal{R}_{n,i}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right) = \sum_{k=1}^{K} \mathcal{R}_{n,i}^{k}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right). \tag{8}$$

E. Validity Function

As we have stated above, vehicles in group $G_{n,i}$ can receive the ith layer data, we define the indicator $e_{n,i}$ to show the validity of the ith layer data received by these vehicles from the perspective of the physical layer,

$$e_{n,i}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right) = \begin{cases} 1, & \text{if } \mathcal{R}_{n,i}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right) \cdot \Delta t \geq \widehat{\mathcal{D}}_i \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\widehat{\mathcal{D}}_i$ is the size of the ith layer data of a GOP. $e_{n,i} = 1$ means that vehicles in group $G_{n,i}$ can receive the ith layer data successfully and $e_{n,i} = 0$ otherwise.

In the process of decoding SVC videos, the ith layer data can be correctly decoded if and only if the data from the 1st layer to the ith layer have been successfully received [15], [16]. Therefore, it is necessary to define a validity function to measure whether the received data is valid or not.

The Validity Function: Let $\Omega_{n,i}^{v}$ denote the effectiveness of the stream data received by vehicle v ($\forall v \in \mathcal{V}_n$) calculated by

$$\Omega_{n,i}^{v}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right) = \prod_{i^*=1}^{i} e_{n,i^*}\left(\boldsymbol{H}_n^t, \boldsymbol{C}_n^t; \boldsymbol{\mathcal{H}}_{-n}^{t}\right). \tag{10}$$

$\Omega^v_{n,i} = 1$ means that vehicle v receives the data from the 1st layer to the ith successfully, i.e., the ith layer data can be decoded; $\Omega^v_{n,i} = 0$ otherwise.

IV. Markov Game-Based Problem Formulation

In this study, we aim to maximize all vehicles' long-term traffic video quality while guaranteeing that each layer data received can be decoded successfully. Considering (10), the optimization problem is formulated as

$$\underset{\boldsymbol{H}^t_1,\ldots,\boldsymbol{H}^t_N;\boldsymbol{C}^t_1,\ldots,\boldsymbol{C}^t_N}{\text{Maximize:}} \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{v=1}^{V_n}\kappa_i\Omega^v_{n,i}\left(\boldsymbol{H}^t_n, \boldsymbol{C}^t_n; \boldsymbol{\mathcal{H}}^t_{-n}\right) \quad (11)$$

subject to (1), (3), and (4)

where κ_i ($\kappa_i \geq 0$) (dollars per vehicle) is the price of the ith layer data, which can be used to balance the fairness and efficiency in terms of video layers. We have the following observations on the problem.

1) The problem (11) is a long-term optimization problem, which needs a sequence of actions to be taken over time. The traditional methods, such as static optimization aim to search the optimal/suboptimal solution to maximize the short-term reward according to the current observation, so they are not feasible approaches.
2) The reward of each UAV depends on not only its own actions, but also other UAVs actions, i.e., there exists complicated interactions among agents.

According to the above observations, the problem (11) is reformulated as a multiagent coordination problem in the form of an MG [43], [44]. The MG is defined by N agents (UAVs) with a tuple $\langle\mathcal{S}, \mathcal{A}, \mathcal{P}, r\rangle$. In the following statement, the terms "agent" and "UAV" are used interchangable. At each time slot, agent n ($\forall n \in \mathcal{N}$) selects an action based on its policy π_n and local observation o_n to maximize its long-term reward, and then, agent n obtains a reward r_n from the environment and local observation of the next time slot o_{n+1}. At the same time, the environment transits into the next state based on the state transition function. It should be noted that all agents take actions synchronously. The details of the MG are as follows.

1) $\mathcal{S}$ is the state feature space of the environment. In this study, $\mathcal{S}$ is made up of the observation from all UAVs, i.e., $\mathcal{S} = \{\mathcal{O}_n|\forall n \in \mathcal{N}\}$, where $\mathcal{O}_n$ is the observation space of UAV n. The space $\mathcal{O}_n$ includes: the UAV's coordinate $[X_n, Y_n, Z_n]$ ($\forall n \in \mathcal{N}$), the vehicle's coordinate at the starting of the tth time slot $[X_{n,v}(t), Y_{n,v}(t), Z_{n,v}(t)]$ ($\forall v \in \mathcal{V}_n$). Let $s = [o_1, \ldots, o_n, \ldots, o_N]$ denote the current state feature of the environment, where o_n is the current observation of UAV n.
2) $\mathcal{A}$ is the action space for all agents that is written as $\mathcal{A} = \{\mathcal{A}_n|\forall n \in \mathcal{N}\}$, where $\mathcal{A}_n = (\boldsymbol{H}^t_n, \boldsymbol{C}^t_n)$ is the action set of UAV n, $\boldsymbol{H}^t_n$ is the PRB allocation strategy, and $\boldsymbol{C}^t_n$ is the vehicle association strategy of UAV n. Let $a = [a_1, \ldots, a_n, \ldots, a_N]$ denote the current action of all UAVs, where a_n is the action of UAV n selected from π_n. To decrease communication overhead of the system, the local observation of UAVs are not interchanged among them, therefore, each UAV learns its own policy based on the local observation π_n: $\mathcal{O}_n \to P(a_n)$, where $P(a_n)$ is the action probability of UAV n.
3) $\mathcal{P}$ is the state transition function. To be specific, the vehicle's coordinate at the next slot is determined by the vehicle's current location, speed, and accelerated speed.
4) r is the immediate reward sets $r = \{r_n|\forall n \in \mathcal{N}\}$, where r_n is the reward of UAV n given by the environment. It is found from (11) that the reward of each UAV depends on not only its own action but also actions of other UAVs, consequently, the immediate reward function of UAV n is a mapping $\mathcal{S}\times\mathcal{A}_1\times\cdots\times\mathcal{A}_N \to r_n$, which is written as

$$r_n = \sum_{i=1}^{I}\sum_{v=1}^{V_n}\kappa_i\Omega^v_{n,i}\left(\boldsymbol{H}^t_n, \boldsymbol{C}^t_n; \boldsymbol{\mathcal{H}}^t_{-n}\right). \quad (12)$$

Centralized reinforcement learning (CRL) can be utilized to solve the MG, where all agents are treated as a single agent. However, since the joint action space of CRL grows exponentially with the number of agents, the scalability of CRL is poor. Independent multiagent reinforcement learning (IMARL), such as independent Q-learning [45], is a simple method to solve the MG. Since each agent takes action independently to maximize its reward while other agents are considered as a part of the environment in IMARL, communication overhead among agents can be ignored. However, IMARL may induce the variance for environment of each agent, which makes the environment nonstationary and non-Markovian. Consequently, the convergence performance cannot be guaranteed.

To overcome the above drawbacks, the study employs a novel multiagent DRL approach, namely, MAAC, to deal with the MG in the next section.

V. Problem Solution With Multiactor Attention Critic Approach

The base idea of MAAC is to centrally train the critic for each agent by selectively paying attention to information from other agents while distributedly training and executing policy in each agent. The framework of the MAAC-based scheme is shown in Fig. 3.

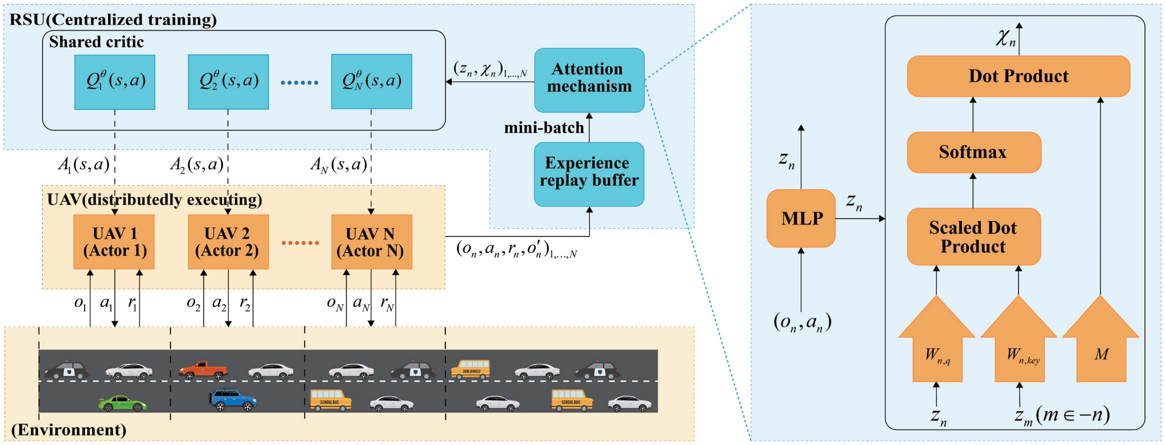
A. Entropy Objective

Similar to our previous work [27], to enhance the exploration ability of the agents, an entropy item $\mathcal{H}(\pi(a|s)) = -\log\pi(a|s)$ is added to the reward. Correspondingly, the objective with the entropy item is named as the entropy objective [46], which is given by

$$J(\pi) = \mathbb{E}\left\{\sum_{t=0}^{\infty}\gamma^t\left[r_t - \lambda\log\pi(a_t|s_t)\right]|\pi\right\} \quad (13)$$

where $\gamma \in [0, 1]$ is the discount factor and λ is the temperature parameter used to adjust the stochasticity of the policy. For the given $\forall s_0$ and a_0, the entropy objective (13) is evolved into the Q-value function

$$Q(s, a) = \mathbb{E}\left\{\sum_{t=0}^{\infty}\gamma^t\left[r_t - \lambda\log\pi(a_t|s_t)\right]|s_0 = s, a_0 = a, \pi\right\}. \quad (14)$$



(a) (b)

Fig. 3. (a) Framework of MAAC-based scheme. (b) Attention mechanism.

B. Critic Part

From (12), it is observed that the reward of agent n is determined not only by agent n's action a_n, but also influenced by other agents' actions denoted by a_{-n}, where $-n$ denotes the set of all agents with the exception of agent n, i.e., $-n = [m|\forall m \in \mathcal{N}, m \neq n]$. Therefore, in MAAC framework shown in Fig. 3, all critics are considered as a joint critic that share the global state s and action a and are trained together by the RSU when evaluating Q-values $Q_n^\theta(s, a)$ $(\forall n \in \mathcal{N})$. The Q-value of agent n is calculated by

$$Q_n^\theta(s, a) = F_n(z_n(o_n, a_n), \chi_n) \tag{15}$$

where F_n is a two-layers multilayer perceptron (MLP) network and $z_n(o_n, a_n)$ is a one-layer MLP network embedding function. χ_n represents the contribution to $Q_n^\theta(s, a)$ from the rest of agents, where $s = (o_n, o_{-n})$ and $a = (a_n, a_{-n})$, respectively. To utilize other agents' information effectively, we attend to selectively pay special attention to certain agents. In this study, an attention mechanism is leveraged to calculate χ_n [47]

$$\chi_n = \sum_{m\in -n} \omega_{n,m} \cdot \text{LeakyReLU}(M \cdot z_m(o_m, a_m)) \tag{16}$$

where $z_m(o_m, a_m)$ is the embedding function of agent m that is linearly transformed by a shared matrix M and activated by a nonlinearity activation function named LeakyReLU$(\cdot)$. $\omega_{n,m}$ is an attention coefficient used to measure the contribution weight of agent m to agent n that is obtained after the following two stages as shown in Fig. 3(b). First, we employ a query-key system with a set of attention hyperparameters $(W_{n,q}, W_{n,\text{key}}, M)$ to calculate the similarity value of embedding functions between agent n and agent m, i.e., $\int_{n,m} = z_m^T W_{n,\text{key}}^T W_{n,q} z_n$, where the linear transformation matrix $W_{n,\text{key}}$ transforms z_m into a "key" while $W_{n,q}$ transforms z_n into a "query," and then, the scaled dot product of the "key" and the "query" is the similarity. Second, the similarity values of each agent m $(\forall m \in -n)$ with agent n are combined as a vector and softmaxed

$$\omega_{n,m} = \frac{\exp\left(z_m^T W_{n,\text{key}}^T W_{n,q} z_n\right)}{\sum_{m\in -n} \exp\left(z_m^T W_{n,\text{key}}^T W_{n,q} z_n\right)} \tag{17}$$

where $\omega_{n,m}$ satisfies $0 \leq \omega_{n,m} \leq 1$ and $\sum_{m\in -n} \omega_{n,m} = 1$.

The shared parameter of Q-value networks θ is renewed by minimizing a joint loss function

$$\text{Loss}(\theta) = \sum_{n=1}^{N} \mathbb{E}_{(s,a,r,s')\sim D}\left[\left(Q_n^\theta(s, a) - y_n\right)^2\right] \tag{18}$$

where $y_n = r_n + \gamma \mathbb{E}_{a'\sim\pi^{\varphi}(s')}[Q_n^{\vartheta}(s', a') - \lambda \log(\pi^{\varphi_n}(a'_n|s'_n))]$ is the target value of agent n in which ϑ and φ are the parameters of target critic and target actors, respectively. θ is renewed by

$$\theta(t+1) \leftarrow \theta(t) - \alpha_{c,t}\nabla_\theta \text{Loss}(\theta) \tag{19}$$

where $\alpha_{c,t}$ is the critics' learning rate. The target critic parameter ϑ is updated by an exponentially weighted moving average of the current critic parameter θ

$$\vartheta(t+1) \leftarrow \rho\vartheta(t) + (1-\rho)\theta(t+1) \tag{20}$$

where ρ satisfies $0 \leq \rho \leq 1$.

C. Actor Part

The target of each actor is to acquire the best parameter ϕ_n to maximize $J(\pi_{\phi_n})$. Policy gradient methods are widely used to solve this kind of optimization problems, which seek for a local maximum in $J(\pi_{\phi_n})$ by ascending the gradient of the policy w.r.t. its arguments ϕ_n

$$\phi_n(t+1) \leftarrow \phi_n(t) + \alpha_{a,t}\nabla_{\phi_n} J\left(\pi_{\phi_n}\right) \tag{21}$$

where $\alpha_{a,t}$ is the actor's learning rate. The policy gradient $\nabla_{\phi_n} J(\pi_{\phi_n})$ is calculated by

$$\begin{aligned}\nabla_{\phi_n} J\left(\pi_{\phi_n}\right) = \mathbb{E}_{(s,a,r,s')\sim D}\big[&\nabla_{\phi_n} \log\left(\pi^{\phi_n}(a_n|o_n)\right) \\ &\left(-\lambda \log\left(\pi^{\phi_n}(a_n|o_n)\right) + A_n(s; a_n, a_{-n})\right)\big]\end{aligned} \tag{22}$$

where $A_n(s; a_n, a_{-n})$ is an advantage function that is used to evaluate whether action a_n can induce an increase in Q-value or whether the increase is caused by other agents' action a_{-n}. This advantage function encourages the actions that directly improve an agent's rewards, i.e.,

$$A_n(s; a_n, a_{-n}) = Q_n^\theta(s; a_n, a_{-n}) - B(s, a_{-n}) \tag{23}$$

where $B(s, a_{-n})$ is a counterfactual baseline function [48] that marginalizes out action a_n from $Q_n^{\theta}(s; a_n, a_{-n})$ while a_{-n} is fixed and is calculated by

$$B(s, a_{-n}) = \mathbb{E}_{a_n \sim \pi_n(o_n)}\big[Q_n^{\theta}(s; a_n, a_{-n})\big]. \tag{24}$$

Since the action space of the MG is discrete, $B(s, a_{-n})$ is given by

$$\begin{aligned} B(s, a_{-n}) &= \mathbb{E}_{a_n \sim \pi_n(o_n)}\big[Q_n^{\theta}(s; a_n, a_{-n})\big] \\ &= \sum_{a_n' \in A_n} \pi\left(a_n' | o_n\right) Q_n^{\theta}(s; a_n, a_{-n}). \end{aligned} \tag{25}$$

Similar with (20), the target policy parameter φ_n is renewed by

$$\varphi_n(t+1) \leftarrow \rho\varphi_n(t) + (1-\rho)\phi_n(t+1). \tag{26}$$

D. MAAC-Based Algorithm

Fig. 3 is the architecture of MAAC-based algorithm, the details of which are given in Algorithm 1. At each step, all agents synchronously interact with the environment based on their own local observation. After a_n is taken, agent n receives the reward r_n and obtains the next local observation o_n'. The samples generated by all agents are combined together as $(o_n, a_n, r_n, o_n')_{1,\ldots,N}$ and stored in the experience replay memory D. After $T_{\min}$ steps, the RSU (critics) randomly samples a minibatch B_c records from D and updates the critic parameter θ by minimizing the joint regression loss function Loss(θ). Each actor trains its policy networks based on the samples drawn from D and the advantage function calculated by the RSU. Finally, the target parameters $\vartheta(t)$ and $\varphi_n(t)(n \in \mathcal{N})$ are updated by employing an exponentially moving average method. The learning rates $\alpha_{c,t}$ and $\alpha_{a,t}$ satisfy [49]

$$\begin{aligned} &\sum_{t=0}^{\infty} \alpha_{c,t} = \infty, \sum_{t=0}^{\infty} \alpha_{c,t}^2 < \infty \\ &\sum_{t=0}^{\infty} \alpha_{a,t} = \infty, \sum_{t=0}^{\infty} \alpha_{a,t}^2 < \infty, \lim_{t\to\infty} \frac{\alpha_{a,t}}{\alpha_{c,t}} = 0. \end{aligned} \tag{27}$$

Algorithm 1: MAAC-Based Algorithm

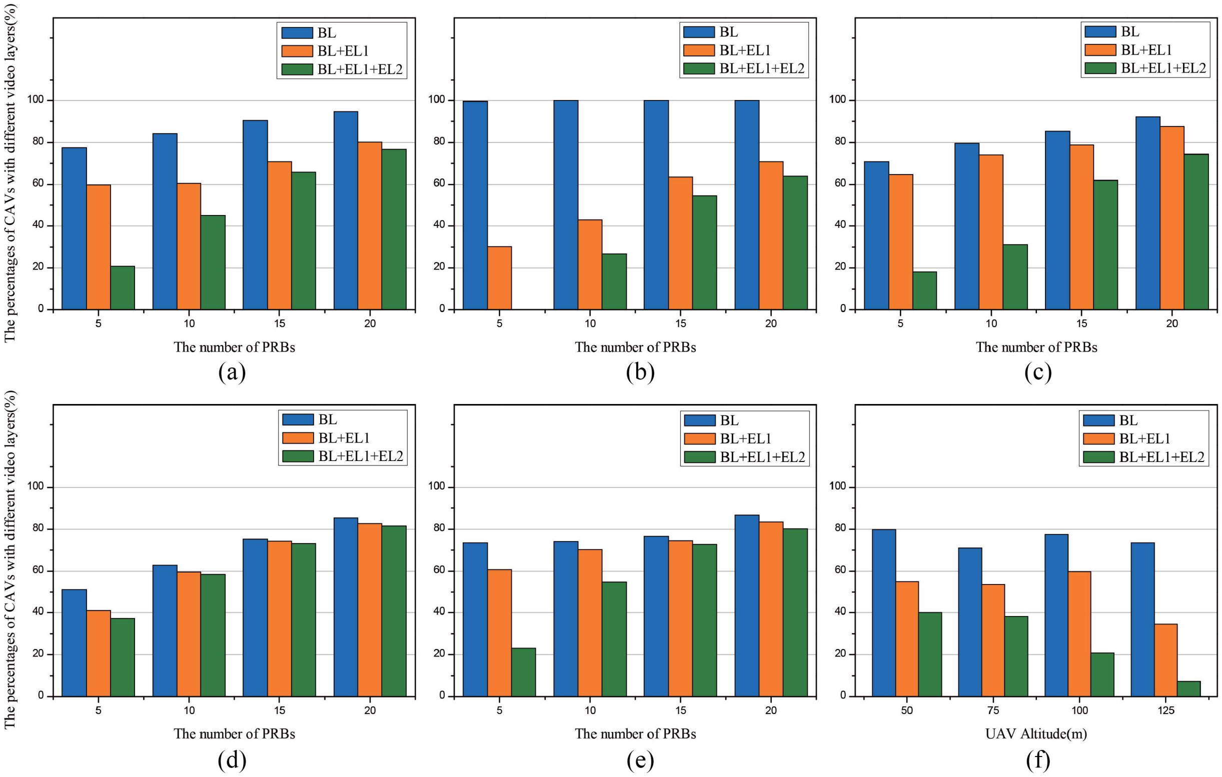
```
begin
    Initialize environment parameters;
    Initialize $T$, $T_{\min}$, $T_c$, and $T_p$;
    Initialize $\theta$, $\vartheta$, $\phi_n$, and $\varphi_n$ $(n \in \mathcal{N})$;
    Initialize experience replay buffer $D$;
    $T_{\text{update}} \leftarrow 0$;
    for each episode do
        // Generate training data by
           agents
        for $t = 1, \cdots, T$ do
            Choose action $a_n \sim \pi_n(o_n)$ for each agent $n$;
            Take action $a_n$, observe $o_n'$ and reward $r_n$ for
            each agent;
            Save the samples of all agents in $D$ by the
            RSU $D \leftarrow (o_n, a_n, r_n, o_n')_{1,\cdots,N} \cup D$;
            $T_{\text{update}} \leftarrow T_{\text{update}} + 1$;
        end
        if $T_{\text{update}} \geq T_{\min}$ then
            // Training critics by the RSU
            for $t_c = 1, \cdots, T_c$ do
                Sample minibatch $B_c \sim D$;
                Update Q-value according to (15);
                Renew $\theta$ according to (19);
            end
            // Training policy by each
               agent
            for $t_p = 1, \cdots, T_p$ do
                Sample minibatch $B_p \sim D$;
                Calculating the advantage function by the
                RSU according to (23) and transmitting
                the result to each agent;
                Each agent updates $\phi_n$ based on (21);
            end
            Update the target parameters according
            to (20) and (26), respectively;
            $T_{\text{update}} \leftarrow 0$
        end
    end
end
```

VI. Simulation Results and Discussions

In the section, simulation results of the MAAC-based algorithm and baseline approaches are presented. The simulations are implemented on a Python-based simulator, where the software environment is TensorFlow 1.15.0 with Python 3.6.5 and the hardware environment is a CPU-based server that has 8 GB 3200 MHz DDR4, 3.0 GHz AMD Core R5, and 512G memory.

In the simulation, vehicles are randomly generated on a 500 m urban road and the trajectories of vehicles are produced by simulation of urban mobility (SUMO). UAVs are deployed right above the road to capture and multicast the traffic videos to vehicles and the hovering altitudes range from 50 to 125 m. An RSU is deployed in the middle of the road that is used to assist UAVs to train the policy. The number of UAVs ranges from 1 to 6. The carrier frequency f_c is 5.9 GHz and the total bandwidth is 20 MHz [41]. The transmission power of UAV n is set as $p_n^k = 20$ mW $(\forall n \in \mathcal{N})$. The length of each time slot $\Delta t = 0.5$ s. The number of video layers is $I = 3$. The values of channel parameters are $\delta = 9.61$, $\mu = 0.16$, $\xi_{\text{LoS}} = 1$dB, and $\xi_{\text{NLoS}} = 20$dB [41]. $\widehat{\mathcal{D}}_1 = 49.3$ Kb, $\widehat{\mathcal{D}}_2 = 79$ Kb, and $\widehat{\mathcal{D}}_3 = 187$ Kb [50]. The learning parameters are set as $\lambda = 5 \times 10^{-3}$ and $\gamma = 0.95$, respectively. The learning rates $\alpha_{a,t}$ and $\alpha_{c,t}$ satisfy (27) and are set by *trial-and-error* [51], [52], [53], which are $\alpha_{a,t} = 3 \times 10^{-3}$ and $\alpha_{c,t} = 3 \times 10^{-2}$ in this article, respectively. The details can be found in the supporting document.

Fig. 4(a)–(f) shows the percentages of vehicles with different video layers w.r.t. the number of PRBs of the MAAC-based algorithm when the video layer price parameters κ_i and UAVs' altitudes H are varying. Fig. 4(a) acts as a baseline, where all layers have equal prices, i.e., $\kappa_1 = \kappa_2 = \kappa_3 = 0.5$. From the



(a) (b) (c)

(d) (e) (f)

Fig. 4. Percentages of vehicles with different video layers w.r.t. the number of PRBs of the MAAC-based algorithm when the video layer price parameters κ_i and UAVs' altitudes H are varying (a) $\kappa_1 = 0.5, \kappa_2 = 0.5, \kappa_3 = 0.5, H = 100$ m. (b) $\kappa_1 = 5, \kappa_2 = 0.5, \kappa_3 = 0.5, H = 100$ m. (c) $\kappa_1 = 0.5, \kappa_2 = 5, \kappa_3 0.5, H = 100$ m. (d) $\kappa_1 = 0.5, \kappa_2 = 0.5, \kappa_3 5, H = 100$ m. (e) $\kappa_1 = 0.5, \kappa_2 = 5, \kappa_3 = 5, H = 100$ m. (f) $\kappa_1 = 0.5, \kappa_2 = 0.5, \kappa_3 = 0.5$, PRBS = 5.

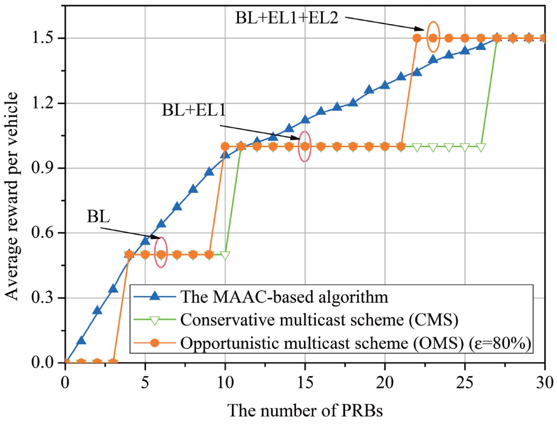


Fig. 5. Performance comparisons among the MAAC-based algorithm and traditional schemes.

figure, it is observed that the video quality has a steady growth with the rising of the number of PRBs. Fig. 4(b)–(e) shows the effectiveness of video layer price parameters κ_i on video quality when the flight height $H = 100$ m. From Fig. 4(b), it is found that for the fixed the number of PRBs, the percentage of vehicles with the BL data is increased to 100% compared with Fig. 4(a) when κ_1 rises from 0.5 to 5, while the percentages of the vehicles with BL+EL1 and BL+EL1+EL2 data are significantly decreased. The similar phenomenons can also be observed from Fig. 4(c)–(e), i.e., the increase of a layer price will induce the rising of the percentage of the corresponding video layer. Therefore, by adjusting the video price parameters, the MAAC-based algorithm can get a balance between fairness and efficiency, which is particularly vital for traffic multicast services in UAV-assisted ITS. Fig. 4(f) shows the percentages of vehicles with different video layers w.r.t. the UAV's altitude when $\kappa_1 = \kappa_2 = \kappa_3 = 0.5$ and the number of PRBs is 5. From the figure, it is found that the video quality decreases when the UAV altitude is rising from 50 to 125 m. That is, because the higher altitude results in worse channel quality and lower transmit rate, consequently, lower quality videos are multicasted.

Fig. 5 shows the average reward per vehicle of the MAAC-based algorithm, CMS, and OMS, in which $\varepsilon = 80\%$ vehicles are selected to be served. Since the multicast rate is bounded by the worst channel user in each group, for the sake of fairness, the above schemes are run under the same simulation environment. From the figure, it is observed that the MAAC-based algorithm has the best performance compared with CMS and OMS, even though OMS has slight advantage in a few cases. There are two reasons for the phenomenon. First, the MAAC-based algorithm tries to provide full coverage for all vehicles while 20% vehicles with bad channel conditions are not served in OMS. Second, the MAAC-based algorithm jointly leverages user grouping and stream splitting techniques, which can help UAVs to multicast multiple version videos to vehicles that are classed into different groups according to channel conditions. Vice versa, from the figure, it is found that the curves of CMS and OMS are stair stepping. The reason is that compared with the MAAC-based algorithm, group splitting and stream splitting techniques are not utilized in both CMS and OMS. Therefore, all vehicles are sharing

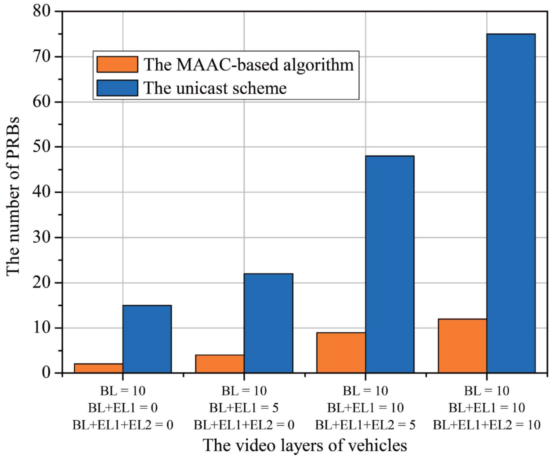


Fig. 6. Consumed PRBs comparisons between the MAAC-based algorithm and the unicast scheme.

the same channel and with the same communication rate, besides, a live video with a sole version is multicasted to all or $\varepsilon = 80\%$ vehicles. Hence, the MAAC-based algorithm has excellent performance in terms of video quality compared with the existing schemes, which indirectly improves the spectrum efficiency.

Fig. 6 shows the consumption of PRBs of the MAAC-based algorithm and the unicast scheme, where $\kappa_1 = \kappa_2 = \kappa_3 = 0.5$, $H = 100$ m, and $V = 10$ in each road segment. From the figure, it is observed that the consumed PRBs of the MAAC-based algorithm is significantly low compared with the unicast scheme. For example, around 15 PRBs are consumed by the unicast scheme while merely 2 PRBs are required in the proposed scheme, when BL = 10, BL + EL1 = 0, and BL + EL1 + EL2 = 0. Besides, it is found that higher quality videos are transmitted, more PRBs are saved by the MAAC-based algorithm in comparison to the unicast scheme. Therefore, the MAAC-based algorithm is an effective method for traffic video transmission services in UAV-assisted ITS when spectrum resource is limited.

To verify the effectiveness of the attention mechanism, the MAAC-based algorithm (Uniform) is also simulated for comparison, which is an ablated version of the proposed MAAC-based algorithm. In this approach, the attention coefficient $\omega_{n,m}$ of (16) is uniformly set as $1/(N-1)$, which can avoid paying more attention on specific agents. The simulation results are shown in Fig. 7. From the figure, we find that the reward of the MAAC-based algorithm nearly keeps constant when the number of agents is increasing, while that of the MAAC-based algorithm (Uniform) decreases significantly when the number is rising. Since each agent has its own individual reward function, i.e., (12), which is tied to other agents' observations and actions, an agent needs to collect information from other agents to calculate its reward. Taking the advantages of the attention mechanism, the MAAC-based algorithm is able to pay more attention to specific agents, which can help to coordinate with other agents more effectively to reduce interference and obtain higher reward than the MAAC-based algorithm (Uniform) does.

Fig. 8 shows convergence and reward performance comparisons among the MAAC-based algorithm, the MAAC-based algorithm (Uniform), independent actor–critic (IAC) [48], centralized actor–critic (CAC) [46], MADDPG, and without learning schemes. The comparisons among these methods also have been listed in Table II. The idea of IAC is derived from independent Q-learning [45] that each agent trains the policy independently to maximize its individual reward while other agents are treated as part of the environment, i.e., no cooperation occurs between different agents. In contrast, all agents

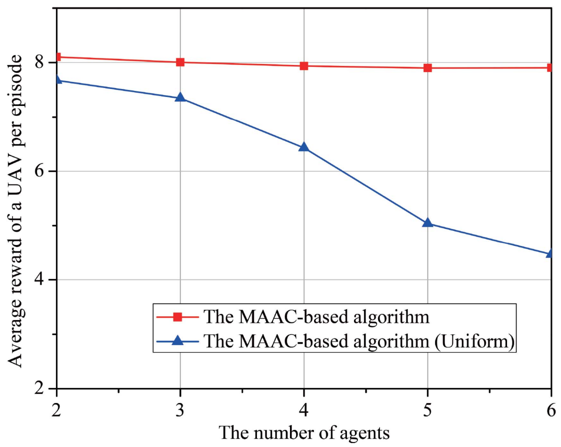


Fig. 7. Performance comparisons between the MAAC-based algorithm and the MAAC-based algorithm (Uniform).

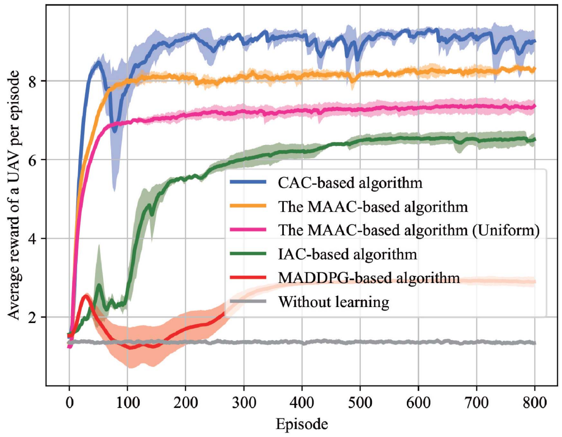


Fig. 8. Performance comparisons among the MAAC-based algorithm, the MAAC-based algorithm (Uniform), CAC, IAC, MADDPG, and the without learning scheme.

TABLE I
ACTION SPACE DIMENSIONS W.R.T. THE NUMBER OF AGENTS (UAVS)

Methods	**The number of UAVs**	**The action space dimensions per actor**
The MAAC-based algorithm	1	1 296
	2	1 296
	3	1 296
	4	1 296
	5	1 296
	6	1 296
The CAC-based algorithm	1	1 296
	2	34 992
	3	944 784
	4	25 509 168
	5	688 747 536
	6	18 596 183 472

TABLE II
COMPARISONS AMONG THE MAAC-BASED ALGORITHM AND OTHER BASELINE METHODS

Methods	**Base algorithm**	**How to cooperate with other agents**	**Number of agents**	**Number of DNNs**	**Input/output dimensions of critic**	**Input/output dimensions of actor**	**Time complexity**[*]
The MAAC-based algorithm	Soft actor-critic	Attention	N	$N+1$	$N \cdot \lvert z_n + \chi_n \rvert / N$	$\lvert o_n \rvert / \lvert a_n \rvert$	$O(\Gamma^2 N \lvert o_n + a_n \rvert)$
The MAAC-based algorithm (Uniform)	Soft actor-critic	Uniform attention	N	$N+1$	$N \cdot \lvert z_n + \chi_n \rvert / N$	$\lvert o_n \rvert / \lvert a_n \rvert$	$O(\Gamma^2 N \lvert o_n + a_n \rvert)$
The MADDPG-based algorithm	MADDPG	Observation and action concatenation	N	$2N$	$N \cdot \lvert o_n + a_n \rvert / 1$	$\lvert o_n \rvert / \lvert a_n \rvert$	$O(\Gamma^2 N \lvert o_n + a_n \rvert^2)$
The IAC-based algorithm	Soft actor-critic	Local observation and action	N	$2N$	$\lvert o_n + a_n \rvert / 1$	$\lvert o_n \rvert / \lvert a_n \rvert$	$O(\Gamma^2 \lvert o_n + a_n \rvert^2)$
The CAC-based algorithm	Soft actor-critic	Global observation and action concatenation	1	2	$N \cdot \lvert o_n + a_n \rvert / 1$	$N \cdot \lvert o_n \rvert / N \cdot \lvert a_n \rvert$	$O(\Gamma^2 N^2 \lvert o_n + a_n \rvert^2)$

[*] Γ is the number of hidden layers and $|\cdot|$ is the cardinality.

are collected and considered as a single agent whose action space is the joint action space of all agents in CAC. This approach can obtain the global optimal policy, however, it is not scalable due to the size of action space increasing exponentially w.r.t. the number of agents, which can be observed from Table I. Thus, in multiagent settings, curse of dimensionality of the action space will make CAC not applicable. In this part, we take the reward of CAC as an upper boundary for comparison when $N = 3$. Besides, to guarantee the fairness, similar with MAAC, both IAC and CAC add entropy term in the objective function [like (13)] to ensure stability and exploration.

In Fig. 8, the darker lines denote the average values and the shaded area is the average values ± the standard error. From the figure, we find that the proposed MAAC-based algorithm has a slight loss on reward compared to CAC. However, CAC suffers higher variance and requires more episodes to reach convergence than the MAAC-based algorithm. The main reason is that the action space size of CAC reaches 944 784 when $N = 3$ (please refer to Table I), it requires more time to explore the optimal policy. On the other hand, in the MAAC-based algorithm, each agent trains its policy based on the local observation and only the advantage function trained by the RSU is transmitted to the agent, which has great advantage of scalability over CAC. It is observed from the figure that IAC suffers higher variance and obtains significantly lower reward than the MAAC-based algorithm. The reason is that IAC treats other agents as a part of the environment, which increases the uncertainty of the state space, besides, there is no cooperation among agents, which results in suboptimal solution. The MAAC-based algorithm has significantly better performance than MADDPG-based algorithm in terms of reward, variance, and learning speeds. Since the critics of MADDPG concatenate states and actions information of all agents into a single input vector for each critic, the input dimensions of the critic networks are extremely high which causes high computing complexity, low learning efficiency, and low scalability. However, the MAAC-based algorithm can pay more attention to specific agents and compress states and actions data into a constant-sized vector by the attention model, which can make the learning more efficient and scalable. The detailed information of the dimensions of input/output and time complexities is shown in Table II.

VII. CONCLUSION AND FUTURE WORKS

This article proposed a real-time traffic video multicasting scheme for UAVs-assisted ITS, which took advantage of stream splitting and group splitting techniques. The study tried to maximize the long-term video quality of the system by jointly optimizing vehicle grouping and PRB allocation strategies. Considering the interactions among the UAVs, the above optimization problem was subsequently modeled as a MG. Thereafter, a novel multiagent DRL algorithm, namely, MAAC, was employed to solve the MG, in which an attention mechanism was leveraged to make the learning process more effective. Finally, extensive simulation results of the MAAC-based algorithm and the baseline algorithms were presented and discussed. The robustness and scalability of the MAAC-based algorithm will be further considered in our future work.

REFERENCES

[1] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, “A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018.

[2] L. Wang, H. Yang, X. Qi, J. Xu, and K. Wu, “iCast: Fine-grained wireless video streaming over Internet of intelligent vehicles,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 111–123, Feb. 2019.

[3] M. A. Khan, W. Ectors, T. Bellemans, D. Janssens, and G. Wets, “Unmanned aerial vehicle–based traffic analysis: Methodological framework for automated multivehicle trajectory extraction,” *Transp. Res. Rec.*, vol. 2626, no. 1, pp. 25–33, 2017.

[4] J. Liang, J. Zhang, V. C. M. Leung, and X. Wu, “Distributed information exchange with low latency for decision making in vehicular fog computing,” *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18166–18181, Oct. 2022.

[5] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, “UAV-enabled intelligent transportation systems for the smart city: Applications and challenges,” *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, Mar. 2017.

[6] A. Gupta, T. Afrin, E. Scully, and N. Yodo, “Advances of UAVs toward future transportation: The state-of-the-art, challenges, and opportunities,” *Future Transp.*, vol. 1, no. 2, pp. 326–350, 2021.

[7] D. Lecompte and F. Gabin, “Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: Overview and rel-11 enhancements,” *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 68–74, Nov. 2012.

[8] R. O. Afolabi, A. Dadlani, and K. Kim, "Multicast scheduling and resource allocation algorithms for OFDMA-based systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 240–254, 1st Quart., 2013.

[9] T.-P. Low, M.-O. Pun, Y.-W. P. Hong, and C.-C. J. Kuo, "Optimized opportunistic multicast scheduling (OMS) over wireless cellular networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 2, pp. 791–801, Feb. 2010.

[10] Z. Yang, L. Cai, Y. Luo, and J. Pan, "Topology-aware modulation and error-correction coding for cooperative networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 379–387, Feb. 2012.

[11] J. Montalban et al., "Multimedia multicast services in 5G networks: Subgrouping and non-orthogonal multiple access techniques," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 91–95, Mar. 2018.

[12] G. Araniti, I. Bisio, M. De Sanctis, F. Rinaldi, and A. Sciarrone, "Joint coding and multicast subgrouping over satellite-eMBMS networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1004–1016, May 2018.

[13] Z. Zhang et al., "Joint user grouping, version selection, and bandwidth allocation for live video multicasting," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 350–365, Jan. 2022.

[14] M. Condoluci, G. Araniti, A. Molinaro, and A. Iera, "Multicast resource allocation enhanced by channel state feedbacks for multiple scalable video coding streams in LTE networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 2907–2921, May 2016.

[15] M. Zhang, H. Lu, F. Wu, and C. W. Chen, "NOMA-based scalable video multicast in mobile networks with statistical channels," *IEEE Trans. Mobile Comput.*, vol. 20, no. 6, pp. 2238–2253, Jun. 2021.

[16] H. Zhu, Y. Cao, T. Jiang, and Q. Zhang, "Scalable NOMA multicast for SVC streams in cellular networks," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6339–6352, Dec. 2018.

[17] M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Leveraging UAVs for coverage in cell-free vehicular networks: A deep reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 9, pp. 2835–2847, Sep. 2021.

[18] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, "UAV-assisted supporting services connectivity in urban VANETs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3944–3951, Apr. 2019.

[19] L. Zhao, K. Yang, Z. Tan, X. Li, S. Sharma, and Z. Liu, "A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3664–3674, Jun. 2021.

[20] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, "UAV-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5155–5167, Aug. 2021.

[21] Z. Su, Y. Wang, Q. Xu, and N. Zhang, "LVBS: Lightweight vehicular blockchain for secure data sharing in disaster rescue," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 19–32, Jan./Feb. 2022.

[22] S. Han, K. Zhu, M. Zhou, and X. Liu, "Joint deployment optimization and flight trajectory planning for UAV assisted IoT data collection: A bilevel optimization approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21492–21504, Nov. 2022.

[23] L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, and G. Jeon, "Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 81–86, May 2019.

[24] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12382–12395, Nov. 2020.

[25] M. A. Khan et al., "Unmanned aerial vehicle-based traffic analysis: A case study to Analyze traffic streams at urban roundabouts," *Procedia Comput. Sci.*, vol. 130, pp. 636–643, Apr. 2018.

[26] J. Du et al., "Resource pricing and allocation in MEC enabled blockchain systems: An A3C deep reinforcement learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 33–44, Jan./Feb. 2022.

[27] F. Fu, Y. Kang, Z. Zhang, F. R. Yu, and T. Wu, "Soft actor–critic DRL for live transcoding and streaming in vehicular fog-computing-enabled IoV," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1308–1321, Feb. 2021.

[28] Z. Zhang et al., "Energy-efficient secure video streaming in UAV-enabled wireless networks: A safe-DQN approach," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1892–1905, Dec. 2021.

[29] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5567–5576.

[30] Z. Dai, Y. Zhang, W. Zhang, X. Luo, and Z. He, "A multi-agent collaborative environment learning method for UAV deployment and resource allocation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 120–130, Feb. 2022.

[31] P. Xiang, H. Shan, M. Wang, Z. Xiang, and Z. Zhu, "Multi-agent RL enables decentralized spectrum access in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10750–10762, Oct. 2021.

[32] Z. Zhang, X. Wen, Z. Li, S. He, W. Jing, and J. Zhao, "QoS-aware energy-efficient power control in two-tier femtocell networks based on *Q*-learning," in *Proc. 21st Int. Conf. Telecommun. (ICT)*, 2014, pp. 313–317.

[33] J. Wang et al., "Cooperative and competitive multi-agent systems: From optimization to games," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 5, pp. 763–783, May 2022.

[34] Z. Zhou and G. Liu, "RoMFAC: A robust mean-field actor-critic reinforcement learning against adversarial perturbations on states," 2022, *arXiv:2205.07229*.

[35] M. Ye, D. Li, Q.-L. Han, and L. Ding, "Distributed Nash equilibrium seeking for general networked games with bounded disturbances," *IEEE/CAA J. Automatica Sinica*, vol. 10, no. 2, pp. 376–387, Feb. 2023.

[36] M. Wang, H. Shi, and C. Wang, "Distributed cooperative learning for discrete-time strict-feedback multi agent systems over directed graphs," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 10, pp. 1831–1844, Oct. 2022.

[37] C. Wang, Y. Wang, Q.-L. Han, and Y. Wu, "MUTS-based cooperative target stalking for a multi-USV system," *IEEE/CAA J. Automatica Sinica*, early access, Nov. 3, 2022, doi: 10.1109/JAS.2022.106007.

[38] J. Wang, Q. Zhang, and D. Zhao, "Highway lane change decision-making via attention-based deep reinforcement learning," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 3, pp. 567–569, Mar. 2022.

[39] Z. Zhang, H. Liu, M. Zhou, and J. Wang, "Solving dynamic traveling salesman problems with deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2119–2132, Apr. 2023.

[40] R. Radhakrishnan, B. Tirouvengadam, and A. Nayak, "Channel quality-based AMC and smart scheduling scheme for SVC video transmission in LTE MBSFN networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2012, pp. 6514–6518.

[41] L. Zhang and N. Ansari, "Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10573–10580, Oct. 2020.

[42] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[43] M. Li et al., "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proc. Web Conf.*, 2019, pp. 983–994.

[44] H. Shi, G. Liu, K. Zhang, Z. Zhou, and J. Wang, "MARL Sim2real transfer: Merging physical reality with digital virtuality in metaverse," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 4, pp. 2107–2117, Apr. 2023.

[45] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.

[46] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1352–1361.

[47] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[48] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 2974–2982.

[49] Z. Zhang, R. Wang, F. R. Yu, F. Fu, and Q. Yan, "QoS aware transcoding for live streaming in edge-clouds aided HetNets: An enhanced actor-critic approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11295–11308, Nov. 2019.

[50] L. Sun, H. Shan, A. Huang, L. Cai, and H. He, "Channel allocation for adaptive video streaming in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 734–747, Jan. 2017.

[51] C. Chen, N. Lu, B. Jiang, and C. Wang, "A risk-averse remaining useful life estimation for predictive maintenance," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 2, pp. 412–422, Feb. 2021.

[52] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[53] P. Zhang, S. Shu, and M. Zhou, "An online fault detection model and strategies based on SVM-grid in clouds," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 445–456, Mar. 2018.

Fang Fu received the Ph.D. degree from the Department of Information and Communication Engineering, Harbin Engineering University, Harbin, China, in 2014.

She is currently a Postdoctoral Fellow under the supervision of Prof. Laurence T. Yang with Hainan University, Haikou, Hainan, China. Her research interests include federated learning, blockchain, incentive mechanism, mobile-edge computing, and deep reinforcement learning.

Dr. Fu received the Best Paper Award at IEEE Globecom'20 and has served as the Technical Program Committee Member of IEEE ICC'21, VTC'21 Fall, ICC'22, and ICC'23. She serves/served as a reviewer for several IEEE journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE INTERNET OF THINGS JOURNAL, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Bin Xue received the B.S. degree from Shanxi University, Taiyuan, China, in 2020, where he is currently pursuing the master's degree with the School of Physics and Electronic Engineering.

His current research interests include distributed machine learning, vehicular networks, and resource allocation.

Lin Cai (Fellow, IEEE) received the M.A.Sc. and Ph.D. degrees (awarded Outstanding Achievement in Graduate Studies) in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively.

Since 2005, she has been with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, where she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things.

Prof. Cai in 2020, was elected as a Member of the Royal Society of Canada's College of New Scholars, Artists, and Scientists, and a 2020 "Star in Computer Networking and Communications" by N2Women. She has co-founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has been elected to serve the IEEE Vehicular Technology Society Board of Governors, from 2019 to 2024, and served its VP Mobile Radio in 2023. She has been a Voting Board Member of IEEE Women in Engineering from 2022 to 2023. She has served as an Associate Editor-in-Chief for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a Steering Committee Member of the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON BIG DATA, and IEEE TRANSACTIONS ON CLOUD COMPUTING, an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON COMMUNICATIONS, and the Distinguished Lecturer of the IEEE VTS Society and the IEEE Communications Society. He is an NSERC E.W.R. Steacie Memorial Fellow and an Engineering Institute of Canada Fellow.

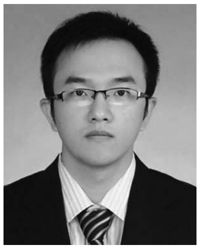
Laurence T. Yang (Fellow, IEEE) received the B.E. degree in computer science and technology and the B.Sc. degree in applied physics from Tsinghua University, Beijing, China, in 1992, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, in 2006.

He is a Professor with the School of Computer Science and Technology, Hainan University, Haikou, China, the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, and the Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. His research has been supported by the National Sciences and Engineering Research Council and the Foundation for Innovation. His research interests include parallel and distributed computing, embedded and ubiquitous computing, and big data.

Zhicai Zhang (Member, IEEE) received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2014.

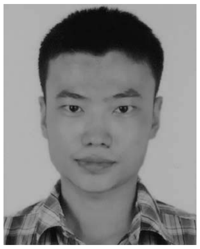
He is currently an Assistant Professor with the School of Computer Science and Technology, Hainan University, Haikou, China. From September 2017 to September 2018, he was with Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. His research interests include edge intelligence, distributed machine learning, and blockchain.

Dr. Zhang received the Best Paper Award at IEEE Globecom'20 and has served as the Technical Program Committee member of IEEE ICC'21, VTC'21 Fall, ICC'22, ICC workshop'22, and ICC'23. He serves/served as a reviewer for several journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and *IEEE Communications Magazine*.



Jia Luo received the Ph.D. degree from Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2020.

He is currently a Lecturer with CQUPT. From April 2018 to April 2019, he was a Visiting Scholar with Carleton University, Ottawa, ON, Canada. His current research interests include UAV communications, mobile-edge computing, and artificial intelligence algorithms.



Chenmeng Wang received the Ph.D. degree in information and telecommunication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2018.

From 2019 to 2021, he was a Postdoctoral Fellow with the University of Alberta, Edmonton, AB, Canada. From 2015 to 2017, he was a visiting Ph.D. student with Carleton University, Ottawa, ON, Canada. He is currently an Associate Professor with Hainan University, Haikou, Hainan, China. His research interests include small-cell HetNets, multi-access edge computing, massive MIMO systems, and resource allocation in mobile networks.