Energy Efficient Computation Offloading in Aerial Edge Networks With Multi-Agent Cooperation

Wenshuai Liu, Bin Li, *Member, IEEE*, Wancheng Xie, Yueyue Dai, *Member, IEEE*,
and Zesong Fei, *Senior Member, IEEE*

Abstract—With the high flexibility of supporting resource-intensive and time-sensitive applications, unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) is proposed as an innovative paradigm to support the mobile users (MUs). As a promising technology, digital twin (DT) is capable of timely mapping the physical entities to virtual models, and reflecting the MEC network state in real-time. In this paper, we first propose an MEC network with multiple movable UAVs and one DT-empowered ground base station to enhance the MEC service for MUs. Considering the limited energy resource of both MUs and UAVs, we formulate an online problem of resource scheduling to minimize the weighted energy consumption of them. To tackle the difficulty of the combinational problem, we formulate it as a Markov decision process (MDP) with multiple types of agents. Since the proposed MDP has huge state space and action space, we propose a deep reinforcement learning approach based on multi-agent proximal policy optimization (MAPPO) with Beta distribution and attention mechanism to pursue the optimal computation offloading policy. Numerical results show that our proposed scheme is able to efficiently reduce the energy consumption and outperforms the benchmarks in performance, convergence speed and utilization of resources.

Index Terms—Mobile edge computing, unmanned aerial vehicle, computation offloading, deep reinforcement learning, digital twin.

I. INTRODUCTION

WITH the rapid advance of Internet of Things (IoT) technology and the extensive deployment of 5G networks,

Wenshuai Liu and Wancheng Xie are with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: liuwenshuai@nuist.edu.cn; zuoyeyiwancheng@gmail.com).

Bin Li is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: bin.li@nuist.edu.cn).

Yueyue Dai is with the School of Cyber Science and Engineering and Research Center of 6G Mobile Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yueyuedai@ieee.org).

Zesong Fei is with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China. (e-mail: feizesong@bit.edu.cn).

the envision and planning of future 6G networks is in progress. It is expected to be empowered with the ability of providing intelligent and green communication features, and realizing the ubiquitous connection between massive objectives [1]–[3]. In order to achieve these requirements, the quality of service and application experience in 6G network need to be significantly enhanced, which involves higher requirements for the computing resources and communication resources of various devices [4]. In this sense, mobile edge computing (MEC) as a popular paradigm naturally arises to alleviate the pressure on resource-constrained devices where the heterogeneous resource-intensive applications are performed. Specifically, the devices can offload their computational tasks to edge servers with powerful computing resources to pursue satisfactory energy consumption and latency [5], [6].

However, the transmission links of terrestrial MEC servers may be influenced by blockage, and the mobile users (MUs) located away from the servers are typically not well covered. Especially in some emergency situations such as disaster area and hot spot area, the fixed MEC servers are intricate to deploy. Recently, unmanned aerial vehicle (UAV) has emerged as a promising technology to assist traditional infrastructure-based MEC networks due to the representative features of high mobility, flexible deployment, and low cost [7], [8]. By cooperatively carrying MEC servers, UAVs can fly closer to the MUs to offload part of the tasks and provide reliable communication links, which is a fast and cost-effective deployment way for computation offloading with low delivery delay and high data rate requirements [9], [10]. The dynamicity of the MEC networks is further enhanced with the participation of UAVs and MUs, and thus motivates the application of online optimizing in MEC service, especially the integration of artificial intelligence approaches that outperforms traditional offline optimization in decision time and adaptability to the environment.

For the learning-based optimizing techniques, the dynamicity in networks imposes strict requirements on accurate state sensing, real-time decision-making, and the precise execution, which presents technical challenges for implementing the UAV-assisted MEC networks [11], [12]. In view of this, as an emerging technology in the 6G era, digital twin (DT) has come into the research vision, which is capable of maintaining the virtual models of physical entities in digital space and monitoring of the overall system status. Currently, DT has been widely used in smart city, healthcare, and intelligent manufacturing [13]. By collecting massive timely data from real space, DT can also realize efficient model training in

dynamic MEC networks, thus providing the MEC services with more intelligent decisions.

A. Related Work

In recent years, UAV-assisted MEC has attracted growing research interests, and extensive research efforts have been conducted to enhance network performance in various scenarios. To exploit the performance of UAV in special situations, Do-Duy *et al.* [14] and Tran *et al.* [15] studied the joint deployment and resource configuration in UAV-aided disaster emergency communications by using iterative methods. To pursue the real-time decision of UAV-assisted networks, Wang *et al.* [16] integrated deep reinforcement learning (DRL) method into iterative convex optimization approaches, and Dai *et al.* [17] studied the minimization of the execution latency in an edge-cloud heterogeneous network comprising UAVs and base stations via DRL approach. For some distributed scenarios, multi-agent reinforcement learning (MARL) emerges as a mighty technique that trains efficient decentralized models, and thereby avoids the dependence of centralized DRL on the whole network state. Wang *et al.* [18] considered the load balance in multi-UAV-assisted MEC under the constraints of users' energy consumption. Peng *et al.* [19] investigated the utilization of MARL to solve the dynamic network resource allocation in UAV-assisted vehicular edge networks. Cai *et al.* [20] studied the data sensing and computation offloading problem in a multi-UAV-assisted MEC, and integrated attention mechanism into MARL algorithm to accelerate the training speed of the network. To further investigate the cooperation of multiple agents, Ji *et al.* [10] exploited the acquisition latency minimization in multi-UAV cellular networks via MARL. Zhao *et al.* [21] investigated the cooperation between multi-UAV and multi-ground MEC servers to minimize the total network utility and enhance the performance of service.

In view of incomparable merits of DT, the idea of combining DT and MEC has made many attempts to build a digital twin edge network (DITEN), but the current research on DITEN is still in its infancy. To fully exploit the capability of DT network for capturing the real-time state of the network, Sun *et al.* [22] investigated the latency minimization problem of DITEN via DRL method and considered the estimated deviation of DT. Sun *et al.* [23] developed an approach to automatically adjust the aggregation frequency of federated learning in DITEN network. Lu *et al.* [24] proposed a blockchain-based federated learning framework that runs in DT for collaborative computing, improving the reliability and security of the system and enhancing data privacy. Liu *et al.* [25] introduced edge collaboration into the DITEN network and realized intelligent offloading of tasks. As a step forward, Do-Duy *et al.* [26] iteratively optimized the joint resources of mobile devices and multiple edge servers to minimize the total latency in industrial IoT networks, and Zhang *et al.* [27] combined DT technology and decomposed MARL into the design of vehicle edge computing networks. Recently, Li *et al.* [28] focused on the DT-aided computation offloading in UAV edge networks where the double deep Q-network algorithm is explored to reduce the system energy consumption.

B. Contributions and Organization

The above-mentioned excellent works have investigated the UAV/DT networks from various aspects. However, the scenarios that involve the joint cooperation among UAVs, base station (BS), and MUs have not been fully exploited. Especially when MUs are deployed in remote or disaster areas, they are not able to directly communicate and synchronize status with BS and DT. Therefore, different from these prior studies, we in this paper propose a real-time distributed optimization framework for the multi-UAV-assisted MEC networks based on MARL, where the policy model can be deployed on MUs and UAVs to make resource allocation decisions, and a DT layer is deployed at the BS to pursue timely monitoring and centralized training, thereby saving the energy of model training on resource-limited MUs and UAVs. Particularly, the main contributions of this paper are presented as follows.

1) We proposed a multi-UAV-assisted MEC network with air-ground cooperation to provide the computing services for the resource-limited MUs, where the interplay between physical environment and DT layer is taken into account. The MUs can partially offload their computational task to UAVs or relay to BS for further computation processing via UAVs. Considering the energy limits of both MUs and UAVs, a weighted energy consumption minimization problem is formulated by jointly designing the association, offloading proportion, trajectory control, and resource allocation on computation and communication.

2) To tackle the high complexity of the formulated real-time problem, we formulate it as a Markov decision process (MDP), and address it by an MARL framework with heterogeneous agents. the high-dimensional hybrid action spaces, the multi-agent proximal policy optimization (MAPPO) algorithm is utilized to train both MUs and UAVs to cooperatively make offloading decisions in dynamic MEC networks.

3) To enhance the performance of training, we adopt Beta distribution and attention mechanism in actor and critic network, respectively. Simulation results show the efficient training convergence and effectiveness of our proposed MAPPO with attention mechanism and Beta distribution (AB-MAPPO) algorithm in optimizing energy consumption. The proposed algorithm outperforms other benchmarks, especially in utilization of computational resource, and flexibility to different deploying scenarios.

The rest of this paper is organized as follows. We first present the system model and formulate the joint problem in Section II. The design of AB-MAPPO algorithm is given in Section III. Section IV presents the simulation results. Finally, we conclude this paper in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first introduce a multi-UAV-assisted MEC system model, including network model, mobility model of MUs, DT model, communication model, and computation model. Then, we formulate the optimization problem to min-
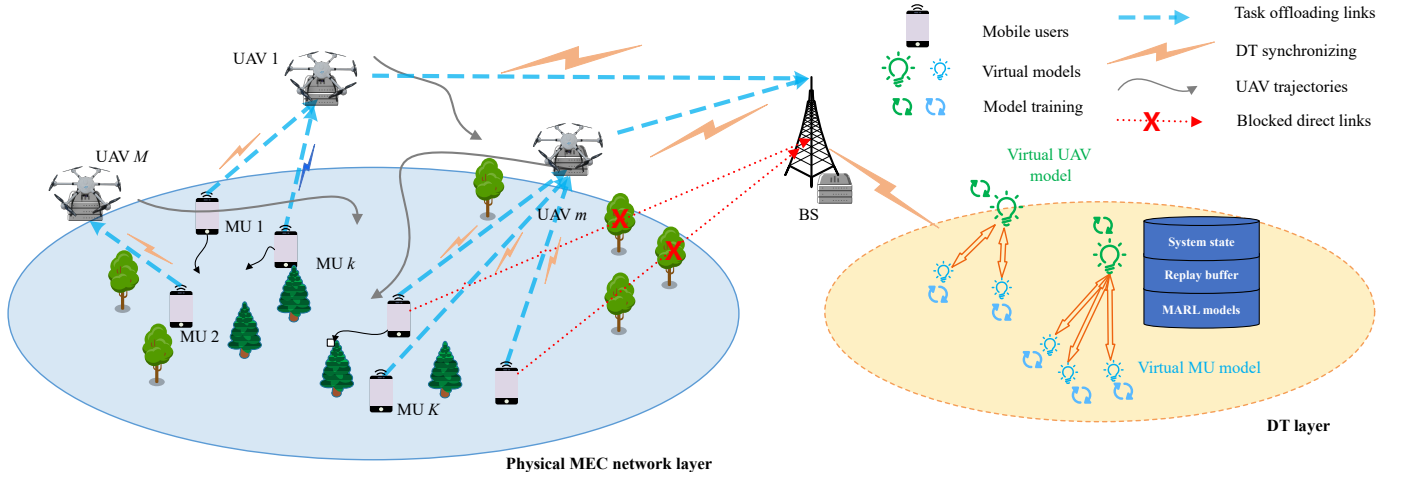
Fig. 1. System model.

imize the weighted energy consumption of MUs and UAVs while ensuring the delay requirement of the tasks.

A. Network Model

As shown in Fig. 1, we consider a multi-UAV-assisted air-ground MEC network containing a BS, M UAVs, and K MUs. For simplicity, we denote the set of MUs by $\mathcal{K} \triangleq \{1, 2, \ldots, K\}$, the set of UAVs by $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$, and the index of BS by $M+1$. The MUs typically have limited battery life and computing capability, which are not capable of completing their resource-intensive tasks efficiently. We assume that UAVs and BS are equipped with MEC servers to provide computing service for MUs. The set of servers is denoted by $\mathcal{M}^\star \triangleq \mathcal{M} \cup \{M+1\}$. Without loss of generality, we introduce a time period T, which is divided into N time slots with time of δ_t. The set of time slots is denoted as $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$. Using three-dimensional Cartesian coordinate system, the locations of MU k, UAV m and BS at time slot n are denoted as $\mathbf{u}_k[n] = [x_k^{\mathrm{MU}}[n], y_k^{\mathrm{MU}}[n], 0]^{\mathrm{T}}$, $\mathbf{q}_m[n] = [x_m^{\mathrm{UAV}}[n], y_m^{\mathrm{UAV}}[n], H]^{\mathrm{T}}$, and $\mathbf{u}_{\mathrm{BS}} = [x_{\mathrm{BS}}, y_{\mathrm{BS}}, H_{\mathrm{B}}]^{\mathrm{T}}$, respectively. Note that the UAVs are flying at a fixed altitude H. To establish an efficient mapping of the physical MEC network, a DT layer is deployed at BS and is equipped with essential functions of data storage, status synchronization and model training. The main notations in this paper are summarized in Table I.

B. Mobility Model of MUs

At the beginning of time slot $n = 1$, all MUs are randomly located, and the MUs are moving according to Gaussian-Markov random model [29]. Considering that each time slot is of short period, the location of MUs are assumed to be static during one time slot. Specifically, at each time slot n, the velocity of $v_k[n]$ and direction $\theta_k[n]$ of MU k are respectively given by

$$v_k[n] = \mu_1 v_k[n-1] + (1-\mu_1)\bar{s} + \sqrt{1-\mu_1^2}\,\Phi_k, \quad (1a)$$

$$\theta_k[n] = \mu_2 \theta_k[n-1] + (1-\mu_2)\bar{\theta} + \sqrt{1-\mu_2^2}\,\Psi_k, \quad (1b)$$

where $0 \leq \mu_1, \mu_2 \leq 1$ are the parameters representing the effect of previous state, $\bar{s}$ and $\bar{\theta}$ are the average velocity and direction of all MUs, respectively. Also, Ψ_k and Φ_k are generated by two independent Gaussian distributions with different mean-variance pairs $(\bar{\xi}_{v_k}, \varsigma_{v_k}^2)$ and $(\bar{\xi}_{\theta_k}, \varsigma_{\theta_k}^2)$ for MU k. Therefore, the coordinate of MUs can be updated as

$$x_k^{\mathrm{MU}}[n] = x_k^{\mathrm{MU}}[n-1] + v_k[n-1]\cos(\theta_k[n-1])\delta_t, \quad (2a)$$

$$y_k^{\mathrm{MU}}[n] = y_k^{\mathrm{MU}}[n-1] + v_k[n-1]\sin(\theta_k[n-1])\delta_t. \quad (2b)$$

C. DT Model of the UAV-assisted MEC Network

In the considered multi-UAV network with MEC service, the DT layer deployed at BS is responsible for state monitoring and virtual twin mapping. To maintain the virtual twins, the physical devices need to upload the DT information of themselves to the DT layer. In this paper, two types of entities are represented, i.e., MUs and the UAVs. To model the virtual twins of them, we focus on their key features which efficiently represent their real-time state corresponding to the optimization scheme.

The virtual twin of each MU k needs to record its location and task information at each time slot n, which is given by

$$\mathrm{DT}_k[n] = \{\mathbf{u}_k[n], \Omega_k[n], \tilde{f}_k^{\mathrm{loc}}[n]\}, \quad (3)$$

where $\Omega_k[n]$ is the task information of users and $\tilde{f}_k^{\mathrm{loc}}[n]$ is the estimated computational frequency for MU k, which will be illustrated in following subsection.

The DT of each UAV should reflect its service status, involving the resource allocation and movement, and is denoted as

$$\mathrm{DT}_m^U[n] = \{\mathbf{q}_m[n], \alpha_{k,m}[n], \tilde{f}_{k,m}[n], \forall k \in \mathcal{K}\}, \quad (4)$$

where $\alpha_{k,m}[n]$ and $\tilde{f}_{k,m}^{\mathrm{edge}}[n]$ are defined the association factor of the network and the estimated computational resource allocated to MU k by UAV m.

TABLE I
MAIN NOTATIONS

Notation	Definition	Notation	Definition
K	The number of MUs	M	The number of UAVs
N	The number of time slots per period	B	The total bandwidth of the network
$\mathcal{M}$	The set of UAVs	$\mathcal{M}^{\star}$	The set of MEC servers
$\mathcal{K}$	Offloading proportion of MU k at time slot n	$\mathcal{N}$	The set of time slot
$\mathbf{q}_m[n]$	The location of MEC server m at time slot n	$\mathbf{u}_k[n]$	The location of MU k at time slot n
$\mathbf{u}_{\text{BS}}$	The location of BS	$L_k[n]$	The size of input data of MU k
$\rho_k[n]$	The offloading proportion of MU k	$T_{k,m}^{\text{ecmp}}[n]$	The actual edge computing time of MU k
$C_k[n]$	Number of CPU cycles required for processing one bit data of MU k at time slot n	$\alpha_{k,m}[n]$	The association variable between MU k and UAV m at time slot n
$f_{k,m}^{\text{edge}}[n]$	The actual allocated CPU frequency of UAV m to MU k at time slot n	$f_k^{\text{loc}}[n]$	The actual CPU frequency of MU k at time slot n
$T_k^{\text{loc}}[n], T_k^{\text{edge}}[n]$	The actual local computing time and edge computing time of MU k	$B_{k,m}[n]$	The bandwidth allocated to link of MU k and UAV m at time slot n
$\boldsymbol{v}_m[n], \mathbf{a}_m[n]$	The acceleration and velocity of UAV m at time slot n	$s(t), o_i(t),$ $a_i(t), r_i(t)$	The global state, observation and reward of agent i at time step t

D. Computation Model

Each MU generates a task $\Omega_k[n] = (L_k[n], C_k[n])$ with a latency requirement of δ_t at the beginning of each time slot n, where $L_k[n]$ and $C_k[n]$ denote the size of input data and average number of central process unit (CPU) cycles for processing one bit data of MU k's task, respectively. The tasks can be divided into two parts at random and executed concurrently, with $(1 - \rho_k[n])L_k[n]$ bits of input data for local computing, and $\rho_k[n]L_k[n]$ bits offloaded to MEC servers associated with BS and UAVs for edge computing. Denoting the offloading association of MU k at time slot n as $\alpha_{k,m}[n]$, we have

$$\alpha_{k,m}[n] \in \{0, 1\}, \forall k \in \mathcal{K}, m \in \mathcal{M}^{\star}, n \in \mathcal{N}. \quad (5)$$

The details are illustrated as follows.

1) Local computing: With $\alpha_{k,0}[n] = 1$, MU k computes total task by itself locally, indicating that $\rho_k[n] = 0$.

According to prior MEC research [17], by adopting dynamic voltage frequency scaling (DVFS) technique, the estimated frequency of MU k at time slot n in DT can be denoted as

$$\tilde{f}_k^{\text{loc}}[n] = \min\{Y_k^{\text{loc}}[n]/\delta_t, f_{\max}^{\text{loc}}\}, \quad (6)$$

where $f_{\max}^{\text{loc}}$ denotes the maximum CPU frequency of MUs, $Y_k^{\text{loc}}[n] = (1 - \rho_k[n])L_k[n]C_k[n]$ denotes the total computing cycles of MU k's task. It means that for energy efficient objective, the task is processed with the minimum CPU frequency required to complete it according to DVFS, thus we have $\tilde{f}_k^{\text{loc}}[n] = Y_k^{\text{loc}}[n]/\delta_t$, and $\tilde{f}_k^{\text{loc}}[n]$ is limited by the maximum CPU frequency $f_{\max}^{\text{loc}}$.

It is noteworthy that the DT layer can't fully reflect the state of MUs and UAVs due to some issues such as the hysteresis of status synchronization and the transmission error. Thus, a deviation is introduced to model the estimated error of DT and can be utilized to verify the robustness of the system. The deviation may occur from computational frequency [22], [25], [26], and location data. Denoting $\hat{f}_k^{\text{loc}}[n]$ as the estimated CPU

frequency of MU k, the estimated local computing time of MU k at time slot n can be expressed as $\tilde{T}_k^{\text{loc}}[n] = Y_k^{\text{loc}}[n]/\tilde{f}_k^{\text{loc}}[n]$. By denoting $\hat{f}_k^{\text{loc}}[n]$ as the estimated deviation of actual frequency $f_k^{\text{loc}}[n] = \tilde{f}_k^{\text{loc}}[n] + \hat{f}_k^{\text{loc}}[n]$, the computing latency gap of MU k between DT and actual value can be expressed as

$$\Delta T_k^{\text{loc}}[n] = \frac{-Y_k^{\text{loc}}[n]\hat{f}_k^{\text{loc}}[n]}{\tilde{f}_k^{\text{loc}}[n](\tilde{f}_k^{\text{loc}}[n] + \hat{f}_k^{\text{loc}}[n])}. \quad (7)$$

As a result, the actual local computing time of MU k can be calculated by

$$T_k^{\text{loc}}[n] = \tilde{T}_k^{\text{loc}}[n] + \Delta T_k^{\text{loc}}[n]. \quad (8)$$

2) Edge computing: MUs can request UAVs for offloading their tasks to MEC servers at UAVs or BSs. The procedures are respectively illustrated as follows.

- **UAV computing**: A part of MU k's task is transmitted to UAV m, and executed by the MEC server on UAV. In this case, we have $\alpha_{k,m}[n] = 1$ and $\alpha_{k,M+1}[n] = 0$.
- **BS computing**: Considering the complex ground environment such as obstacle blocking, the MU-BS links are too poor to support transmission. Therefore, a certain part of task is first transmitted to UAV m, and further relayed to BS for executing. To this end, we have $\alpha_{k,m}[n] = \alpha_{k,M+1}[n] = 1$.

We adopt probabilistic line-of-sight (LoS) channel to represent UAV-MU links. The probability of geometrical LoS between the UAV and MUs depends on the environment status and the elevation angle. Denoting the elevation angle as $\vartheta_k[n]$, and the LoS probability of MU k and UAV m at time slot n as $\mathbb{P}(\text{LoS}, \vartheta_k[n])$, which is approximated to following form

$$\mathbb{P}(\text{LoS}, \vartheta_k[n]) = \frac{1}{1 + a\exp(-b(\vartheta_k[n] - a))}, \quad (9)$$

where a and b are the parameters related to environment, and $\vartheta_k[n]$ is given by

$$\vartheta_k[n] = \frac{180}{\pi} \arctan\left(\frac{H}{\|\mathbf{u}_k[n] - \mathbf{q}_m[n]\|}\right). \quad (10)$$

Additionally, the non-line-of-sight (NLoS) channel probability can be expressed as $\mathbb{P}(\text{NLoS}, \vartheta_k[n]) = 1 - \mathbb{P}(\text{LoS}, \vartheta_k[n])$. Therefore, the expected channel power gain of from MU k to UAV m is given by

$$h_{k,m}[n] = \frac{\beta_0 \mathbb{P}(\text{LoS}, \vartheta_k[n])}{\|\mathbf{u}_k[n] - \mathbf{q}_m[n]\|^{\tilde{\iota}}} + \frac{\nu \beta_0 \mathbb{P}(\text{NLoS}, \vartheta_k[n])}{\|\mathbf{u}_k[n] - \mathbf{q}_m[n]\|^{\tilde{\iota}}}$$
$$= \frac{\nu \beta_0 \mathbb{P}(\text{NLoS}, \vartheta_k[n])}{\|\mathbf{u}_k[n] - \mathbf{q}_m[n]\|^{\tilde{\iota}}}, \tag{11}$$

where $\tilde{\iota}$ is the path loss exponent, ν is the NLoS attenuation, β_0 is the channel power gain at the reference distance of 1 m. Similar to [30], we assume that the change of LoS probability $\mathbb{P}(\text{LoS}, \vartheta_k[n])$ in each time slot is negligible. Furthermore, channel from UAV m to BS can be modeled by quasi-static block fading LoS link [31], i.e.,

$$h_m^{\text{rel}}[n] = \frac{\beta_0}{\|\mathbf{q}_m[n] - \mathbf{u}_{\text{BS}}\|^2}. \tag{12}$$

We consider the orthogonal frequency division multiple access scheme for data transmission. In each time slot, MU k first requests for offloading, then UAV m allocates an orthogonal frequency bandwidth of $B_{k,m}[n]$ for transmission. Therefore, the transmit rates between MU k and UAVs, and UAV-BS are given by

$$R_k[n] = \sum_{m=1}^{M} \alpha_{k,m}[n] B_{k,m}[n] \log_2 \left(1 + \frac{p_k h_{k,m}[n]}{B_{k,m}[n] N_0}\right), \tag{13}$$

$$R_m^{\text{rel}}[n] = B_u \log_2 \left(1 + \frac{p_m h_m^{\text{rel}}[n]}{B_u N_0}\right), \tag{14}$$

where p_k and p_m denote the uplink transmit power of MU k and UAV m, respectively, B_u is the bandwidth allocated to UAVs for relaying, and N_0 is the noise power density.

Since each MU can associate with at most one UAV or BS in each time slot n, and the bandwidth should be only allocated to links between associated devices, we have

$$\sum_{m=0}^{M} \alpha_{k,m}[n] = 1, \forall k \in \mathcal{K}, n \in \mathcal{N}, \tag{15}$$

$$\sum_{m=1}^{M} \sum_{k=1}^{K} B_{k,m}[n] \leq B, \forall n \in \mathcal{N}, \tag{16}$$

$$B_{k,m}[n] \geq 0, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{17}$$

$$\lceil B_{k,m}[n]/B \rceil = \alpha_{k,m}[n], \forall k \in \mathcal{K}, m \in \mathcal{M}, n \in \mathcal{N}, \tag{18}$$

where B denotes the available channel bandwidth. Then the size of total task transmitted to UAV m by MUs and the CPU cycles of MU k's task executed by UAV m can be respectively calculated as

$$L_m^{\text{trans}}[n] = \sum_{k=1}^{K} \alpha_{k,m}[n] \rho_k[n] L_k[n], \tag{19}$$

$$Y_{k,m}^{\text{edge}}[n] = \alpha_{k,m}[n](1 - \alpha_{k,M+1}[n]) \rho_k[n] L_k[n] C_k[n]. \tag{20}$$

Following prior research [17], [32], we assume that the computation result with small size can be downloaded to MU with a negligible transmission latency. Additionally, BS is equipped with energy supply and powerful MEC server, hence the computing time and energy at BS are not considered. Thus, the total estimated edge computing latency of MU k can be expressed as

$$T_k^{\text{edge}}[n] = \sum_{m=1}^{M} \alpha_{k,m}[n] \left(T_k^{\text{off}}[n] + \alpha_{k,M+1}[n] T_{k,m}^{\text{rel}}[n] \right. $$
$$\left. + (1 - \alpha_{k,M+1}[n]) T_{k,m}^{\text{ecmp}}[n]\right), \tag{21}$$

where $T_k^{\text{off}}[n] = L_k[n] \rho_k[n]/R_{k,m}[n]$ denotes the transmission delay of MU k, $T_m^{\text{rel}}[n] = L_k[n] \rho_k[n]/R_{k,m}^{\text{rel}}[n]$ denotes the time of UAV m for relaying the task to BS, and $\tilde{f}_{k,m}^{\text{edge}}[n]$ is the estimated CPU frequency of server m allocated to MU k at time slot n. Accordingly, the computing latency gap of UAV m between DT and actual value can be expressed as

$$\Delta T_{k,m}^{\text{ecmp}}[n] = \frac{-Y_{k,m}^{\text{edge}}[n] \hat{f}_{k,m}^{\text{edge}}[n]}{\tilde{f}_{k,m}^{\text{edge}}[n](\tilde{f}_{k,m}^{\text{edge}}[n] + \hat{f}_{k,m}^{\text{edge}}[n])}, \tag{22}$$

where $\hat{f}_{k,m}^{\text{edge}}[n]$ is the estimated deviation of $f_{k,m}^{\text{edge}}[n]$. Denoting $\tilde{T}_{k,m}^{\text{ecmp}}[n] = Y_{k,m}^{\text{edge}}[n]/\tilde{f}_{k,m}^{\text{edge}}[n]$ as the estimated time of UAV m for computing the task of MU k, the actual edge computing time of MU k at each time slot n can be calculated by

$$T_{k,m}^{\text{ecmp}}[n] = \tilde{T}_{k,m}^{\text{ecmp}}[n] + \Delta T_{k,m}^{\text{ecmp}}[n], \tag{23}$$

and the actual CPU frequency allocated to MU k by UAV m is given by $f_{k,m}^{\text{edge}}[n] = \tilde{f}_{k,m}^{\text{edge}}[n] + \hat{f}_{k,m}^{\text{edge}}[n]$. Since it is inappropriate to allocate the CPU frequency to MUs which are not choosing to offload to the UAVs, i.e., computing at local or BS, we have

$$\lceil \tilde{f}_{k,m}^{\text{edge}}[n]/f_{\max}^{\text{edge}} \rceil = \alpha_{k,m}[n](1 - \alpha_{k,M+1}[n]),$$
$$\forall k \in \mathcal{K}, m \in \mathcal{M}, n \in \mathcal{N}. \tag{24}$$

For the location data, a random noise is imposed on the observation of devices, and thus has an influence on the decisions of algorithm, which will illustrated in Section III.

E. Energy Consumption Model

As the entities (MUs and UAVs) can upload their essential status information to DT layer via UAVs with specific bandwidth, the energy consumption can be evaluated by the DT layer. The energy models of MUs and UAVs are illustrated as follows.

1) MU energy consumption: The energy consumption of MU k comprises the local computing energy and transmit energy, which are denoted as $E_k^{\text{loc}}[n]$ and $E_k^{\text{off}}[n]$, respectively. Therefore, we have

$$E_k^{\text{loc}}[n] = \kappa f_k^{\text{loc}}[n]^2 Y_k^{\text{loc}}[n], \tag{25}$$

$$E_k^{\text{off}}[n] = p_k[n] T_k^{\text{off}}[n], \tag{26}$$

where κ is the effective capacitance coefficient determined by chip architecture.

2) UAV energy consumption: The energy consumption of UAVs is imposed by flying and computation. In each time slot n, UAV m is assumed to fly under the limits of the maximum speed $v_{\max}$ and the maximum acceleration $a_{\max}$, which indicates that

$$\|\mathbf{a}_m[n]\| = \frac{\|\mathbf{v}_m[n+1] - \mathbf{v}_m[n]\|}{\delta_t} \leq a_{\max}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{27}$$

$$\mathbf{q}_m[n+1] = \mathbf{q}_m[n] + \mathbf{v}_m[n]\delta_t + \frac{1}{2}\mathbf{a}_m[n]\delta_t^2, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{28}$$

$$\|\mathbf{v}_m[n]\| = \frac{\|\mathbf{q}_m[n+1] - \mathbf{q}_m[n]\|}{\delta_t} \leq v_{\max}, \forall m \in \mathcal{M}, n \in \mathcal{N}. \tag{29}$$

Subsequently, we adopt the UAV's propulsion energy model introduced by [33], the flying energy of UAV m is then expressed as follows

$$E_m^{\text{fly}}[n] = \left[\frac{1}{2} d_0 \rho s A \|\mathbf{v}_m[n]\|^3 + P_0 \left(1 + \frac{3\|\mathbf{v}_m[n]\|^3}{U_{\text{tip}}^2} \right) \right. \\ \left. + P_i \left(\sqrt{1 + \frac{\|\mathbf{v}_m[n]\|^4}{4v_0^4}} - \frac{\|\mathbf{v}_m[n]\|^2}{2v_0^2} \right) \right] \delta_t, \tag{30}$$

where P_0 and P_i are the blade profile power and induced power in hovering status, respectively, U_{tip} is the tip speed of the rotor blade, v_0 is the mean rotor velocity, d_0 is fuselage drag ratio, s denotes the rotor solidity, ρ denotes the air density, and A denotes the rotor disc area.

Note that the communication energy of UAV is relatively small compared to the energy consumption imposed by the intensive computation and flying, which can be neglected. The computation energy of UAV m is calculated as

$$E_m^{\text{edge}} = \sum_{k=1}^{K} \kappa \alpha_{k,m}[n] f_{k,m}^{\text{edge}}[n]^2 Y_{k,m}^{\text{edge}}[n]. \tag{31}$$

As a consequence, the overall energy consumption of MUs and UAVs are calculated by

$$E_k^{\text{MU}}[n] = E_k^{\text{loc}}[n] + E_k^{\text{off}}[n], \tag{32}$$

$$E_m^{\text{UAV}}[n] = E_m^{\text{fly}}[n] + E_m^{\text{edge}}[n]. \tag{33}$$

F. Problem Formulation

Considering the deficient energy resource of MUs and limited energy budget of UAVs, we pursue a multi-UAV-assisted MEC network with air-ground cooperation design, aiming at minimizing the weighted energy consumption of MUs and UAVs with latency sensitive tasks, by jointly optimizing the MU association $\mathbf{A} \triangleq \{\alpha_{k,m}[n], \forall n \in \mathcal{N}, k \in \mathcal{K}, m \in \mathcal{M}^\star \cup \{0\}\}$, the offloading proportion $\boldsymbol{\varrho} \triangleq \{\rho_k[n], \forall n \in \mathcal{N}, k \in \mathcal{K}\}$, the CPU frequency allocation $\mathbf{F} \triangleq \{\tilde{f}_{k,m}^{\text{edge}}[n], \forall m \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{N}\}$, the bandwidth allocation $\mathbf{B} \triangleq \{B_{k,m}[n], \forall m \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{N}\}$, and the UAV velocity $\mathbf{V} \triangleq \{\mathbf{v}_m[n], \forall m \in \mathcal{M}\}$.

$\mathcal{M}, n \in \mathcal{N}\}$. Therefore, the weighted energy minimization problem is formulated as

$$\mathcal{P}1 \min_{\mathbf{A}, \mathbf{B}, \mathbf{F}, \boldsymbol{\varrho}, \mathbf{V}} \quad \varpi \sum_{n=1}^{N} \sum_{m=1}^{M} E_m^{\text{UAV}}[n] + \sum_{n=1}^{N} \sum_{k=1}^{K} E_k^{\text{MU}}[n] \tag{34a}$$

$$\text{s.t} \quad (5), (15), (16), (17), (18), (24), (27), (28), (29), \tag{34b}$$

$$0 \leq \rho_k[n] \leq 1, \forall k \in \mathcal{K}, n \in \mathcal{N}, \tag{34c}$$

$$\lceil \rho_k[n] \rceil = 1 - \alpha_{k,0}[n], \forall k \in \mathcal{K}, n \in \mathcal{N}, \tag{34d}$$

$$\sum_{k=1}^{K} f_{k,m}^{\text{edge}}[n] \leq f_{\max}^{\text{edge}}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{34e}$$

$$T_k^{\text{loc}}[n] \leq \delta_t, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{34f}$$

$$T_k^{\text{edge}}[n] \leq \delta_t, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{34g}$$

$$\|\mathbf{q}_i[n] - \mathbf{q}_j[n]\|^2 \geq d_{\min}^2, \forall i, j \in \mathcal{M}, i \neq j, n \in \mathcal{N}, \tag{34h}$$

where ϖ is the weight factor, $d_{\min}$ denotes the minimum safe distance between the UAVs. Constraints (5) and (15) ensure the feasibility of the association status, constraints (16)-(18) guarantee that the bandwidth is only allocated to valid links, constraints (27)-(29) denote the velocity and acceleration limits of UAVs, constraints (34c) and (34d) indicate the range and the validness of offloading proportion, constraints (24) and (34e) denotes feasibility of computational resource allocation, constraints (34f) and (34g) denote the computational latency constraints, and constraint (34h) limits the minimum safe distance between UAVs.

It can be readily derived that $\mathcal{P}1$ is a mixed integer nonconvex and combinational problem with huge amount of highly decoupled variables. Furthermore, the proposed problem needs to be timely solved in each time slot n due to the randomly generated tasks and the movement of devices. Nonetheless, it is very intricate to solve the problem by typical iterative techniques such as alternative optimization and genetic algorithm. In the following, we propose an efficient algorithm for this complicated problem by harvesting the MARL approach.

III. PROPOSED DRL APPROACH: AB-MAPPO

In this section, the essential elements of MDP in multi-agent DRL are first illustrated. Then, we endeavor to develop an MAPPO approach with attention mechanism and Beta distribution to address problem $\mathcal{P}1$.

A. MDP Elements Formulation

In MARL setup, problem $\mathcal{P}1$ can be modeled as an MDP with the set of agents $\mathcal{I} \triangleq \{1, \ldots, K+M\}$, state space $\mathcal{S}$, action space $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_{K+M}$. The agents can enhance their policy by interacting with the environment in discrete steps. In each step t, each agent i obtains the current observation $o_i(t)$ from the global environment state $s(t) \triangleq \{o_i(t), \forall i \in \mathcal{I}\}$, takes action $a_i(t) \in \mathcal{A}_i$, then obtains a reward $r_i(t)$, and the environment transfers to a new state $s(t+1)$.

The MDP involve two types of agents, i.e., MUs and UAVs. At the beginning of each step t, the MUs generate actions,

and then request for offloading according to the actions. Afterwards, the UAVs obtain the actions and serve the MUs. The status information is periodically synchronized to DT if is required, and rewards are evaluated centrally by DT. The formulation of MDP elements are illustrated as follows.

1) MDP elements of MUs: The MDP elements of MUs involve the observation $o_k(t)$, action $a_k(t)$ and reward $r_k(t)$, which are presented as follows.

- **Observation:** From the perspective of privacy, MUs can only obtain the positions of themselves $\mathbf{u}_k[n]$ by Global Positioning System, and the positions of MEC servers $\mathbf{q}_m[n]$ from the broadcast of the DT layer. The position data accordingly has a random deviation, which is also imposed to the observations of agents. Furthermore, the task information $\Omega_k[n]$ can be observed. As the task load on UAVs is highly relevant to MUs' offloading decisions, the previous computation load of UAVs are also involved. Therefore, the observation of MU k at step t can be expressed as

$$o_k(t) = \left\{ \mathbf{u}_k[n], \mathbf{q}_m[n], \Omega_k[n], Y_{k,m}^{\text{edge}}[n-1], \forall m \in \mathcal{M} \right\}. \tag{35}$$

- **Action:** The decisions of MUs involve offloading association $\mathbf{A}$, and offloading proportion ϱ. Therefore, for each MU k, the action is decomposed by

$$a_k(t) = \left\{ \alpha_{k,m}[n], \rho_k[n], \forall m \in \mathcal{M}^* \cup \{0\} \right\}. \tag{36}$$

- **Reward:** It can be noticed from (34a) that the MUs need to consider their own influence on the total weighted energy consumption and the load on UAV servers. Hence, the reward of each MU k should involve the energy consumption of both MU k itself and the UAV associated by MU k. The reward of each MU agent k is given by

$$r_k(t) = - \sum_{m=1}^{M} \alpha_{k,m}[n] \left(\varpi E_m^{\text{UAV}}[n] + P^t(T_k^{\text{loc}}[n], T_m^{\text{edge}}[n]) \right) - E_k^{\text{MU}}[n], \tag{37}$$

where

$$P^t(T_k^{\text{loc}}[n], T_m^{\text{edge}}[n]) = \frac{\mu_t}{\delta_t} \left(\text{ReLU}(T_k^{\text{loc}}[n] - \delta_t) + \text{ReLU}(T_m^{\text{edge}}[n] - \delta_t) \right) \tag{38}$$

denotes the penalty for unsatisfaction of latency constraints, μ_t is a penalty factor, and the ReLU($\cdot$) is the rectified linear unit function.

2) MDP element of UAVs: Note that each UAV needs to make decision after MUs give their association. Herein, MDP elements are denoted as follows

- **Observation:** The UAV agent m observes the locations of themselves, locations of all MUs as well as the UAVs, and task information from associated MUs. Denoting $-m$ as the index set of UAVs except for UAV m, we have

$$o_{K+m}(t) = \left\{ \rho_k[n], \Omega_k[n], \mathbf{u}_k[n], \mathbf{q}_m[n], \mathbf{q}_{-m}[n] \right\}. \tag{39}$$

- **Action:** After receiving the requests, UAVs need to allocate the bandwidth, configure the computational frequency, and adjust velocity according to the observations. Therefore, the action of UAV agent m is given by

$$a_{K+m}(t) = \left\{ B_{k,m}[n], \tilde{f}_{k,m}^{\text{edge}}[n], \mathbf{a}_m[n], \forall k \in \mathcal{K} \right\}. \tag{40}$$

- **Reward:** After receiving, relaying and executing the tasks from MUs, the UAVs get rewards from environments. The reward of each UAV m needs to consider the energy consumption of both itself and the served MUs, i.e.,

$$r_{K+m}(t) = - \sum_{k=1}^{K} \alpha_{k,m}[n] \left(E_k^{\text{MU}}[n] + P^t(T_k^{\text{loc}}[n], T_m^{\text{edge}}[n]) \right) + \varpi E_m^{\text{UAV}}[n] + P^o(\mathbf{q}_m[n]) + P^c(\mathbf{q}_m[n]) + P^d(\mathbf{q}_m[n]), \tag{41}$$

where

$$P^o(\mathbf{q}_m[n]) = \mu_o \|\mathbf{q}_m[n] - \text{clip}(\mathbf{q}_m[n], 0, W)\|, \tag{42}$$

denotes the penalty when UAVs try to fly out of the square boundary with width W, and μ_o is a penalty factor.

$$P^d(\mathbf{q}_m[n]) = \frac{1}{W} \left(\left\| \mathbf{q}_m[n] - \frac{1}{|\mathcal{K}_m|} \sum_{k=1}^{K} \alpha_{k,m}[n] \mathbf{w}_k[n] \right\| - d_{\text{th}} \right) \tag{43}$$

guides the distance from UAV m to MUs, d_{th} is a distance threshold, and $|\mathcal{K}_m|$ is the set of MUs associated with UAV m. In addition,

$$P^c(\mathbf{q}_m[n])$$
$$= \mu_c \sum_{j=1, j \neq m}^{M} \min \left\{ \|\mathbf{q}_m[n] - \mathbf{q}_j[n]\| - d_{\min}, 0 \right\} / d_{\min} \tag{44}$$

is the penalty for disobeying the safety distance $d_{\min}$ between UAVs, and μ_c is corresponding penalty factor. It is assumed that the UAVs will stop at the boundary if they try to fly out of it, and thus $\mathbf{q}_m[n] \leftarrow \text{clip}(\mathbf{q}_m[n-1] + \mathbf{v}_m[n-1]\delta_t + \frac{1}{2}\mathbf{a}_m[n-1]\delta_t^2, 0, W)$.

B. MAPPO-based DRL Approach With CTDE

As a variant of PPO specialized for multi-agent settings, MAPPO is one of the state-of-the-art MARL algorithms [34]. Based on on-policy scheme, each agent has an actor, a critic, and a replay buffer. As DT is capable of evaluating the global state of the environment, the centralized training and decentralized executing (CTDE) scheme can be adopted. In MAPPO with CTDE, the critics evaluate the centralized state-value function of global environment state. Specifically, the agents repeatedly input the observations $o_i(t)$ into actor networks and get $a_i(t)$ and $r_i(t)$ for an episode, storing the experiences into buffer. Then in the end of an episode, the agents update their policies. Firstly, they sample some batches of experiences $\{o_i(t), a_i(t), r_i(t), s(t), \text{pr}_i(t)\}, \forall i \in \mathcal{I}$ from
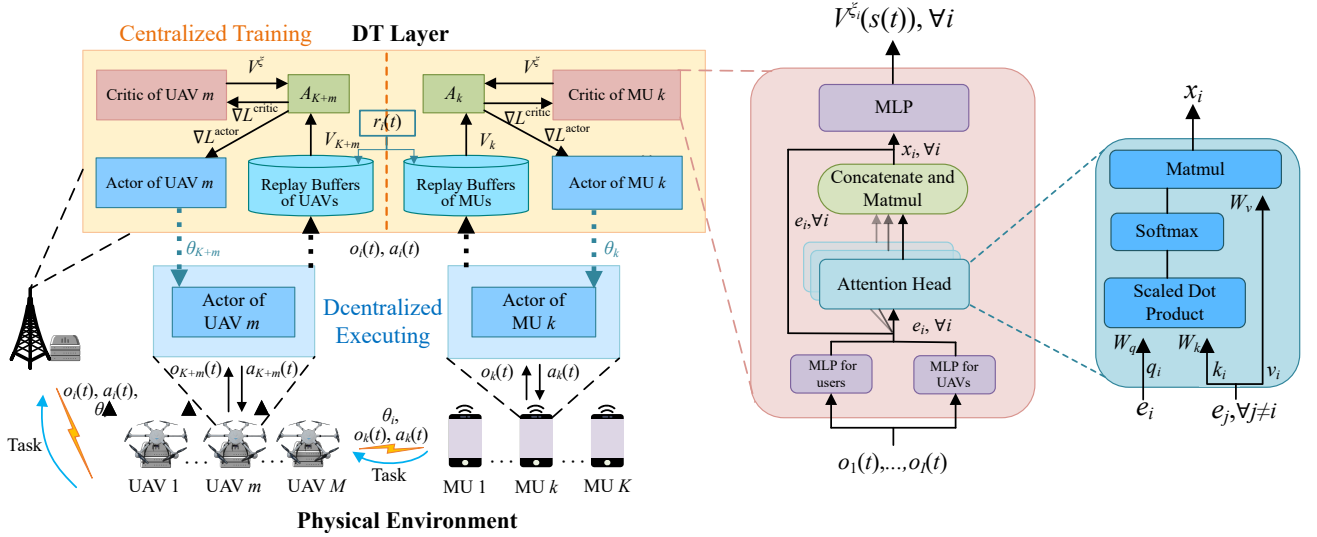
Fig. 2. Training framework of AB-MAPPO.

their buffers, where the $\text{pr}_i(t)$ denotes the log-probability for sampling the action $a_i(t)$. Then, in each update, the actor and critic update the parameters with the loss of policy and global state-value, respectively. The loss of actor i is calculated by

$$
\begin{aligned}
L^{\text{actor}}(\theta_i) = \mathbb{E}_{\pi_{\theta_i}} &\left\{ \min\left[\frac{\pi_{\theta_i}(a_i(t)|o_i(t))}{\pi_{\theta_i'}(a_i(t)|o_i(t))} \hat{A}_i(t), \right.\right. \\
&\left.\left. \text{clip}\left(\frac{\pi_{\theta_i}(a_i(t)|o_i(t))}{\pi_{\theta_i'}(a_i(t)|o_i(t))}, 1-\epsilon, 1+\epsilon \right) \hat{A}_i(t) \right] \right\},
\end{aligned}
\tag{45}
$$

where $\pi_{\theta_i'}$ and π_{θ_i} denote the old and current policy, respectively. $\hat{A}_i(t)$ is the estimation of advantage function $A_i(t) = Q_i(s(t), a_i(t)) - V_i(s(t))$. In PPO series, the generalized advantage estimation (GAE) is adopted to improve the performance, which is defined as follows

$$
\begin{aligned}
\hat{A}_i(t) = \sum_{l=0}^{\infty} (\gamma\lambda)^l &\Big(R_i(t+l) + \gamma V_i\big(s(t+1+l)\big) \\
&- V_i(s(t+l)) \Big),
\end{aligned}
\tag{46}
$$

where γ is the discount factor, λ is the parameter of GAE for bias-variance tradeoff in estimation, and $V_i(s(t)) = \sum_{l=0}^{\infty} \gamma^l R_i(t+l)$ is the cumulative discounted reward, which also represents the state-value function. Denoting $V^{\xi_i}(s(t))$ as the state-value function estimated by the critic of agent i, the loss of critic i is given by

$$
L_i^{\text{critic}}(\xi_i) = \frac{1}{2}\Big[V^{\xi_i}\big(s(t)\big) - V_i\big(s(t)\big) \Big]^2,
\tag{47}
$$

where ξ_i is the parameter of i-th critic network. Therefore, the actor and critic can be updated according to (45) and (47), respectively.

It is worth noting that the constraints need to be satisfied by action-remapping as follows. First, the output vectors of actor networks in MAPPO are sampled from distributions and are divided according to the sequence for concatenating

the variables of actions. Then, the divided vectors are scaled into their origin domain. Furthermore, some constraints need to be tackled by normalization or rounding operation. For each MU k, the index of its associated UAV is with respect to the maximum value of the divided vector corresponding to $\{\alpha_{k,m}[n], \forall m \in \mathcal{M}^* \cup \{0\}\}$, and thus $\{\alpha_{k,m}[n], \forall m \in \mathcal{M} \cup \{0\}\}$ is obtained. If $\alpha_{k,0}[n] = 0$, $\alpha_{k,M+1}[n]$ will be decided by rounding its entry in the divided vector. For constraints (18) and (34d) on $B_{k,m}[n]$ and $\rho_k[n]$, we multiply their divided vectors by $[\alpha_{k,1}[n], \alpha_{k,2}[n], \ldots, \alpha_{k,M}[n]]$ and $1 - \alpha_{k,0}[n]$ as action masks. And then the constraints (16) and (34e) are satisfied by normalizing their divided vectors.

C. MAPPO-based Training Framework

As displayed in Fig. 2, during the proposed DT-assisted CTDE process, the MUs and UAVs perform computation offloading according to the actions given by the actor networks of their agents in physical environment, send their own experiences and synchronizing system status to the DT layer. Then, the DT layer evaluates the global environment state by the observations of agents, updates the buffers, and gets the predicted values. After updating the actors and critics, the parameters of actor networks are downloaded to UAVs and MUs. Note that the network parameters can be shared between homogeneous agents [34]. To fully exploit the performance of MAPPO approach, we introduce Beta distribution and attention, which are stated as below.

1) Beta policy: The above-mentioned actions are typically continuous and bounded, such as $\varrho \in [0, 1]$ and $\mathbf{a}_k[n] \in [-a_{\max}, a_{\max}]$. However, conventional action sampling from Gaussian distribution in policy networks will unavoidably introduce an estimation bias of policy gradient, since the boundary effects will be imposed by force clipping the values of out-of-bound actions. To tackle this problem, we adopt Beta distribution instead of Gaussian distribution in terms of the

---

Algorithm 1 Proposed AB-MAPPO algorithm

1: Initialize $n = 1$, episode length epl, PPO epochs Pe, and maximum episodes $e^{\max}$.
2: **for** agent $i \in \mathcal{I}$ **do**
3:   Initialize actor networks θ_i, critic networks ξ_i, replay buffer $\boldsymbol{B}_i$;
4: **end for**
5: **for** Episode = 1,..., $e^{\max}$ **do**
6:   **for** $t = 1,...,$ epl **do**
7:     Obtain $o_k(t), \forall k \in \mathcal{K}$ from the environment;
8:     Execute the action $a_k(t), \forall k \in \mathcal{K}$;
9:     Obtain $o_{K+m}(t), \forall m \in \mathcal{M}$ from the environment;
10:     Execute the action $a_{K+m}(t), \forall m \in \mathcal{M}$;
11:     Calculate log-probability $\mathrm{pr}_i(t), \forall i \in \mathcal{I}$;
12:     Synchronize status information to the DT layer if is required;
13:     **if** $n \bmod N = 0$ **then**
14:       Synchronize observations $o_i(t)$, actions $a_i(t)$ to the DT layer;
15:       Evaluate reward $r_i(t), \forall i \in \mathcal{I}$ in DT layer;
16:       Store transition
         $\mathrm{Tr}_i(t) = \{o_i(t), a_i(t), r_i(t), s(t), \mathrm{pr}_i(t)\}, \forall i \in \mathcal{I}$ into buffer B_i;
17:     **end if**
18:     Update $n = n \bmod N + 1$
19:   **end for**
20:   **for** epoch = $1, \ldots$ Pe **do**
21:     **for** agents $i \in \mathcal{I}$ **do**
22:       Update actor θ_i and critic ξ_i according to (45) and (47) by $\forall \mathrm{Tr}_i(t) \in \boldsymbol{B}_i$;
23:     **end for**
24:   **end for**
25:   Download the actor networks from DT layer to UAVs and MUs;
26: **end for**

---

output of policy networks, which has the following form

$$f(x, \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} \left(1 - x\right)^{\beta-1}, \qquad (48)$$

where α and β are the parameters of Beta distribution. Since (48) has also a bounded domain, it is appropriate to sample bounded actions. Moreover, the Beta distribution can be more close to uniform distribution than Gaussian by initializing, and thus the agents can more abundantly explore the action space at initial stage of training.

2) Attention mechanism: With the increasing number of agents, the ever-increasing dimensions of environment state make it difficult for the critic network with a simple fully-connected layer to tackle the input, thereby leading to slower or even difficult convergence of critic network and being harmful to the behavior of actor network. However, for each agent, other agents may impose the impact with different intensities on the state-value, and thus the distinguished attention should be paid to them, which involves the nearby devices and the devices with higher computational load. Therefore, we introduce the multi-head attention unit before the multi-layer

perceptron (MLP) of critic networks to improve the training process. The vectors of observations for two types of agents are first passed through their corresponding 3-layer MLPs to get feature values e_i. Then, all the feature values of agents are sent to the attention heads to get the attention values x_i by

$$\alpha_{i,j} = \mathrm{Softmax}\left(\frac{e_j^{\mathrm{T}} W_{\mathrm{key}}^{\mathrm{T}} W_q e_i}{\sqrt{d_{\mathrm{key}}}}\right), \qquad (49)$$

$$x_i = \sum_{j \neq i} \alpha_{i,j} W_v e_j, \qquad (50)$$

where e_j is the feature value of another agent j, d_{key} is the variance of $e_j^{\mathrm{T}} W_{\mathrm{key}}^{\mathrm{T}} W_q e_i$. The matrix W_{key} transforms e_j into a key, the matrix W_v transforms e_j into a value, and the matrix W_q transforms e_i into a query. Finally, x_i and $o_i(t)$ are sent to the MLP to get the estimated state-value $V^{\xi_i}(s(t))$.

D. Complexity Analysis

Based on above-mentioned discussions, we propose the AB-MAPPO algorithm, which is summarized in Algorithm 1. The complexity of attention module is $\mathcal{O}(I^2 V)$, where V is the length of feature-value vectors, according to [35]. For an MLP, the computational complexity of the j-th layer is $\mathcal{O}(Z_{j-1} Z_j + Z_j Z_{j+1})$, where Z_j is the number of neurons for j-th layer. Hence, the computational complexity of a J-layer MLP is calculated by $\mathcal{O}\left(\sum_{j=2}^{J-1} Z_{j-1} Z_j + Z_j Z_{j+1}\right)$. The actor networks are all 3-layer MLPs, and each of two critic networks for MU and UAV agents have two 3-layer MLPs for the observations of different types agents and one 3-layer MLP after attention module for the output of value function. Therefore, the overall computational complexity for training is calculated by the sum of complexity imposed by actor and critic networks $\mathcal{O}\left(e^{\max}(Pe I^2 V + epl \sum_{j=2}^{J-1} Z_{j-1} Z_j + Z_j Z_{j+1})\right)$, and for one-step execution is just calculated by $\mathcal{O}\left(\sum_{j=2}^{J-1} Z_{j-1} Z_j + Z_j Z_{j+1}\right)$.

IV. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed AB-MAPPO algorithm. The simulation settings and numerical results are presented as follows.

A. Simulation Settings

In our simulations, a 1000×1000 m rectangular area is considered, where the MUs are randomly distributed in region with $x, y \in [0, 1000]$ m, the BS is set at (-500 m, 0 m, 10 m), and the UAVs are flying at $H = 200$ m. Note that since the energy consumption of UAVs and MUs are sensitive and the BS typically has abundant power supply, the energy consumption of the BS involving computation and DT construction is not considered. Other parameter settings of simulation and hyperparameters of MAPPO are summarized in Table II and Table III, according to prior works [16], [17], [33], [36].

The baseline algorithms for comparison with our proposed AB-MAPPO are listed as follows:

TABLE II
ENVIRONMENT PARAMETERS

Parameters	Value
Channel bandwidth B	50 MHz
Channel power gain β	-30 dB
Environment parameters a and b	15,0.5
Time period T	60 s
Time slot δ_t	1 s
Transmit power of MUs p_k^{MU}	0.2 W
Transmit power of UAVs p_m^{UAV}	0.5 W
Path loss exponent $\tilde{\iota}$	2.2
NLoS attenuation ν	0.2
Noise power density N_0	-127 dBm/Hz
Effective capacitance coefficient κ	10^{-27}
Maximum CPU frequency of MU k $f_{\max}^{\text{loc}}$	1 GHz
Maximum CPU frequency of UAV m $f_{\max}^{\text{edge}}$	10 GHz
Maximum velocity of UAV $v_{\max}$	30 m/s
Maximum acceleration $a_{\max}$	5 m/s^2
UAV settings P_0, P_i	39.04 W, 79.07 W
UAV settings U_{tip}, v_0, A	120 m/s, 3.6 m/s, 0.5030 m^2
Weight factor ϖ	0.001

TABLE III
HYPERPARAMETERS PARAMETERS OF AB-MAPPO

Parameters	Value
Total number of steps	80k
Length of an episode epl	300
Learning rate of actor	0.0003
Discount factor of UAVs	0.95
Discount factor of MUs	0.8
Penalty factors μ_o, μ_t, and μ_c	0.1
Number of heads in attention unit	4
Number of mini-batches	1
PPO epoch Pe	5
Learning rate of critic	0.0003
Number of hidden layers of MLPs	1
Hidden size of MLPs	128
Optimizer	Adam
d_{th}	300 m

- **B-MAPPO:** MAPPO without attention mechanism and with Beta distribution.
- **AG-MAPPO:** MAPPO with Gaussian distribution and attention mechanism.
- **MADDPG:** MADDPG is an off-policy algorithm with determinate policy [18]. The exploration noise is set as 0.5, the buffer size is set as 20000, and other parameters are the same to the proposed scheme.
- **Randomized:** The MUs and UAVs execute randomly generated actions by the algorithm.

B. Convergence of the MARL Training Algorithm

We first evaluate the convergence performance of the proposed AB-MAPPO algorithm in Fig. 3(a) and Fig. 3(b). The average episode reward of all MUs is evaluated with comparison of RL baselines under $M = 10$ UAVs and $K = 60$

(a) The average reward of MUs.
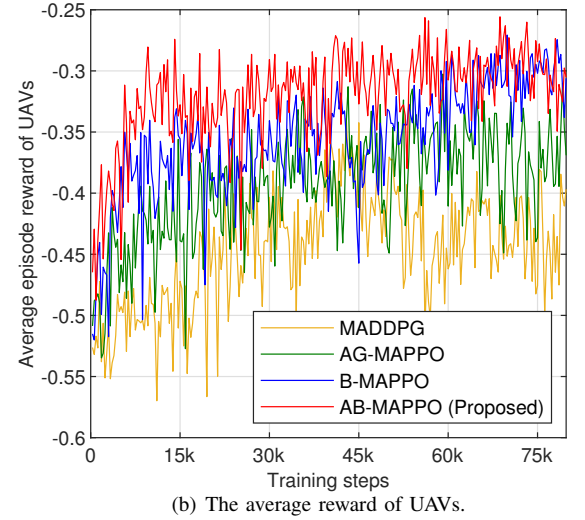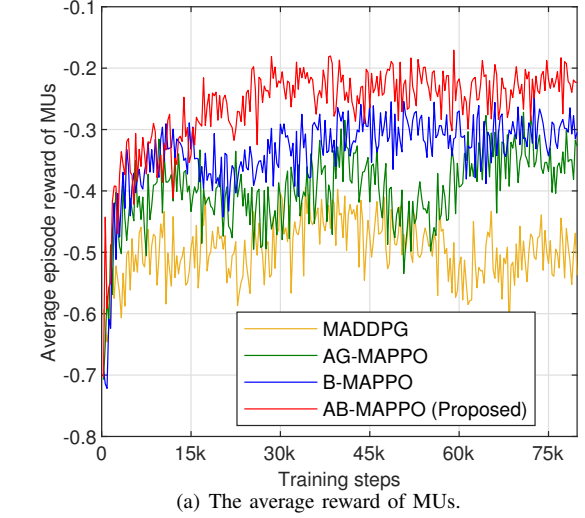

(b) The average reward of UAVs.

Fig. 3. The convergence of the rewards of agents.

MUs. It can be seen that the average episode reward of UAVs obviously increases during the training process and then reaches to the stable values. Intuitively, the proposed AB-MAPPO scheme, B-MAPPO scheme, AG-MAPPO scheme, and MADDPG scheme converge at around 30k, 45k, 60k and 60k steps, respectively. As expected, the proposed scheme outperforms other baselines, with the fastest convergence and reward. Specifically, the proposed AB-MAPPO scheme converge faster than the B-MAPPPO scheme, and the MAPPO schemes with Beta distribution achieve greater rewards than the AG-MAPPO scheme. The reason can be explained as follows: 1) The attention mechanism makes the critic networks quickly concentrate on the significant parts of the global state, thereby accelerating the convergence. 2) With the Beta distribution, the agents explore more uniformly at the initial stage of training, retain better exploration ability, and converge to better solutions, which also presents the superiority of Beta distribution over Gaussian distribution and the MADDPG's exploration noise.

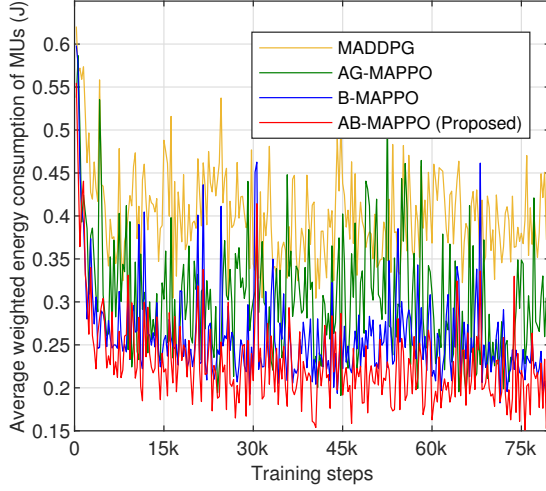Subsequently, Fig. 4 presents the convergence of the average

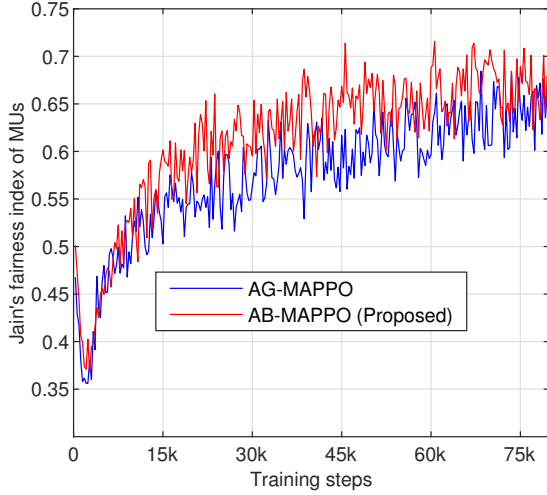Fig. 4. The convergence of weighted energy consumption.

Fig. 6. The impact of MUs on average weighted energy consumption of MUs.

Fig. 5. The improvement of fairness of MUs.
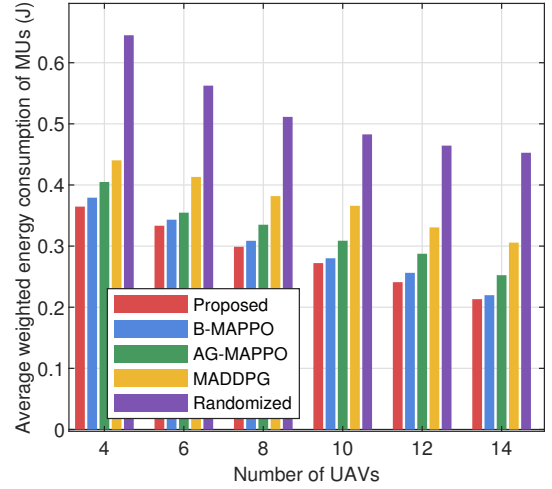
Fig. 7. The impact of number of UAVs on average weighted energy consumption of MUs.

weighted energy consumption of users with $K = 60$ MUs and $M = 10$ UAVs. It indicates that the energy consumption has also been optimized with the increase of reward during the training process. Intuitively, the proposed AB-MAPPO scheme achieves lower energy consumption and has the most rapid convergence. In terms of stability, the energy consumption of the proposed AB-MAPPO scheme goes down more smoothly, the AG-MAPPO and the B-MAPPO schemes have higher fluctuation during training, and the MADDPG scheme is more tortuous. The above results verify the effectiveness and reliability of our proposed scheme.

To verify the fairness of service, we present the varying of Jain's fairness index of MUs in Fig. 5 during training. It is worth noting that the Jain's fairness index of service is denoted as $\frac{\sum\limits_{k=1}^{K} E_k^{\mathrm{MU}}[n])^2}{(K \sum\limits_{k=1}^{K} (E_k^{\mathrm{MU}}[n])^2}$. We can find that the fairness of MUs gradually grows during the training process and the fairness of proposed AB-MAPPO is higher and more stable
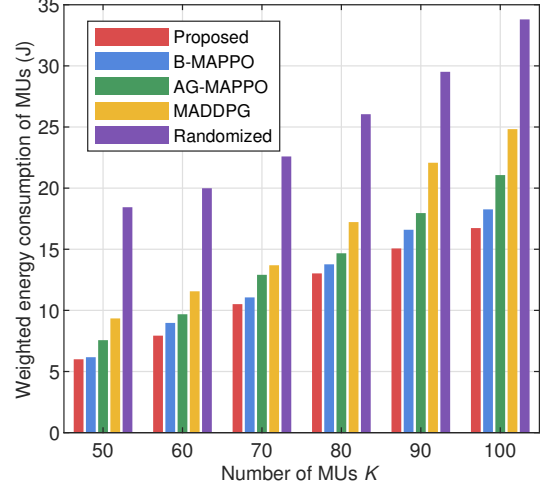
against the AG-MAPPO scheme. Another observation is that the fairness of both schemes slightly decreases at first and then ascends, while the AG-MAPPO scheme converges faster but has lower fairness. The ascending of fairness is due to the fact that the policy of MUs gradually becomes stable, indicating that both the quality and reliability of service have been improved.

C. Comparison of Benchmarks Under Different Settings

To verify the impact of number of MUs on the energy expense of the network, we then evaluate the average weighted energy consumption of MUs, i.e., $\sum\limits_{k=1}^{K} \left(\varpi \sum\limits_{m=1}^{M} \alpha_{k,m}[n] E_m^{\mathrm{UAV}}[n] + E_k^{\mathrm{loc}}[n] \right) / K$, versus the number of MUs in Fig. 6 with $M = 10$ UAVs. From this figure, we observe that the energy cost gradually increases as the number of MUs grows, the proposed AB-MAPPO scheme outperforms the baselines. Additionally, the gaps between
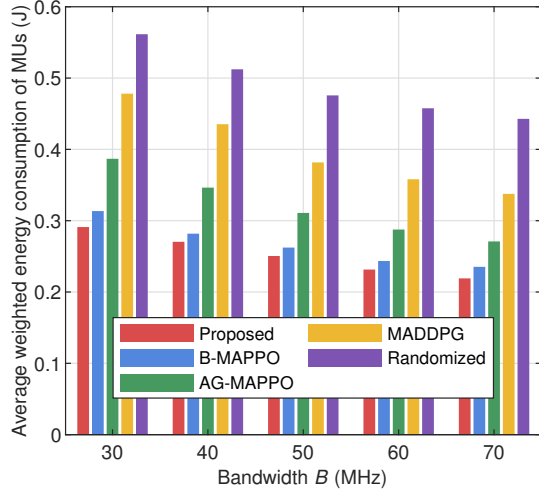
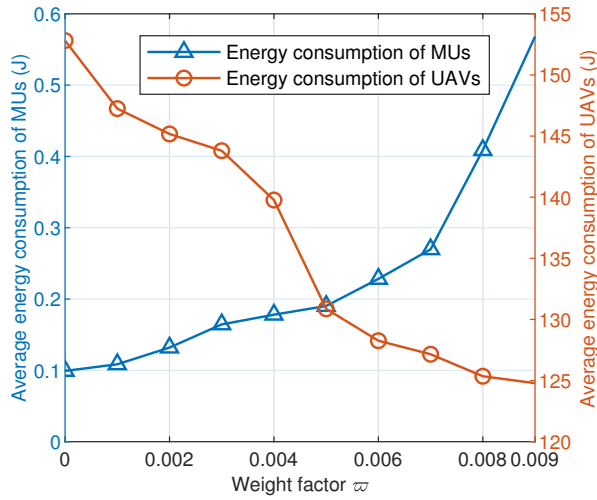Fig. 8. The impact of bandwidth on energy consumption of MUs.

Fig. 9. The impact of weight factor on energy consumption of MUs and UAVs.

Fig. 10. The impact of computational resource.

Fig. 11. The impact of DT deviation.

RL-based schemes have the tendency of becoming larger, the randomized scheme keeps at the highest energy consumption, and the performance gain of attention mechanism increases as the number of MUs grows. Since the average communication resource declines as the number of MUs grows, the latency and energy for transmission of MUs increase, and the MEC servers can process less amount of tasks. Therefore, the MUs tend to computing locally, thereby leading to the ascending of the energy for local computing. Furthermore, the randomized scheme uniformly generates actions, and thus has poor ability to appropriately utilize the abundant resource at BS and UAVs.

Fig. 7 presents the comparison of the proposed scheme and baselines on average weighted energy consumption of $K = 70$ MUs under different number of UAVs. As shown in this figure, when more UAVs participate in the service, the average weighted energy of MUs gradually declines. It can be seen that the proposed AB-MAPPO scheme has the lowest energy cost, and the B-MAPPO scheme is slightly higher
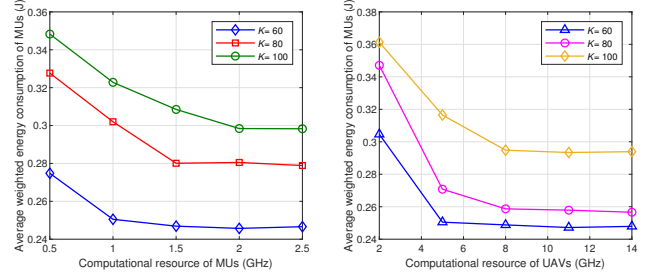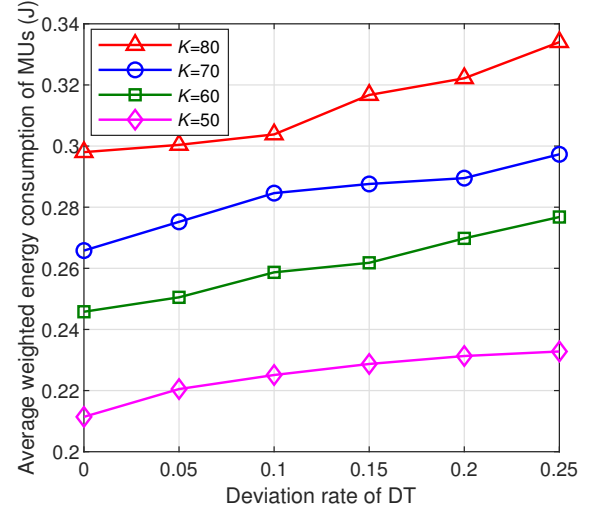
than it. The likely reason is that the Beta distribution can efficiently improve the exploration of UAVs in large action space. Moreover, the MADDPG scheme converges to worse solutions than that of the MAPPO scheme, which indicates that the algorithm is relatively difficult to tackle the increasingly challenging actions.

To gain further insights, Fig. 8 depicts the average weighted energy consumption of MUs versus the channel bandwidth B. It can be seen that the energy consumption gradually decreases when the bandwidth increases. Furthermore, the proposed AB-MAPPO scheme obtains the best performance, and when the bandwidth becomes larger, the gaps between RL schemes have the tendency of decreasing. This is because more abundant communication resource enables the MUs to offload more tasks to UAVs and BS, since lager bandwidth imposes looser constraints on the optimization problem, which may introduce larger penalty to hinder the exploration of agents. This in turn verifies that the proposed AB-MAPPO scheme outperforms the baselines in tackling the problems with strong constraints.

D. The Impact of Environment Settings on the Performance

In Fig. 9, we examine the impact of weight factor ϖ on the optimized energy consumption of UAVs and MUs, under $K = 60$ MUs and $M = 10$ UAVs. It can be observed that as the weight factor ϖ increases from 0 to 0.009, the energy consumption of UAVs sharply decreases at first, and then
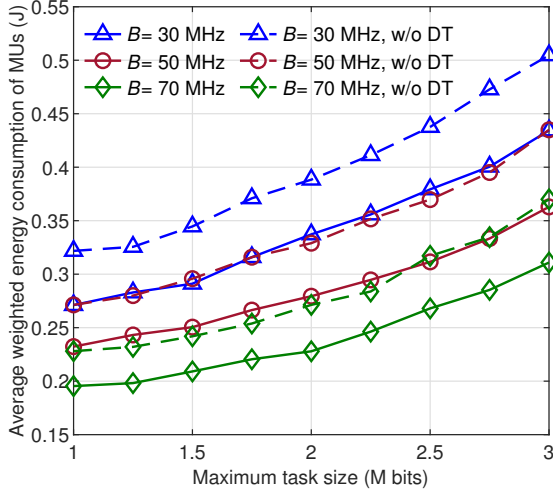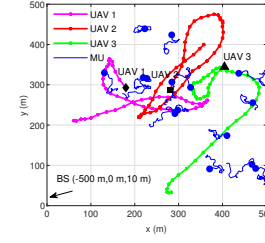
Fig. 12. The impact of maximum task size of users.

(a) Scenario 1 with $K = 12$, $M = 3$, and $T = 50$ s

(b) Scenario 2 with $K = 12$, $M = 3$, and $T = 50$ s

(c) Scenario 3 with $K = 60$, $M = 10$, and $T = 60$ s

(d) Scenario 4 with $K = 60$, $M = 10$, and $T = 60$ s

Fig. 13. The trajectories of UAVs with different scenarios.

becomes smoother. Meanwhile, the energy of MUs increases sharply. The reason for this trend is that as ϖ increases, the energy consumption of UAVs is more emphasized and more tasks are computed locally by MUs, which means that more computing energy is consumed by MUs and less energy is consumed by UAVs.

To evaluate the impact of computational resource on the energy expense of users, we present the average weighted energy consumption of users versus the maximum computational frequency of users and UAVs in Fig. 10(a) and Fig. 10(b), respectively. It can be readily observed that as the computational frequency of users and UAVs increases, the energy consumption gradually decreases, and keeps at about constant values. The likely reason is that the increase of computational resource leads to an increase of satisfaction with computational service. Hence, the policy of MUs on offloading proportion is accordingly adjusted to balance the energy consumption and the penalty for the improvement of reward. Then, the abundant computational resource makes the latency requirements satisfied, and the policies become stable.

We then evaluate the deviation of DT on system performance in Fig. 11. We can find that as the deviation rate of DT increases, the average energy consumption of MUs has the tendency of increasing. This can be explained by the fact that the DT deviation reduces the accuracy of observations and the actions of agents. As such, the improvement of the policies for agents is influenced, and thus the reward and the energy consumption become worse. Another observation is that as the deviation of DT increases from 0 to 0.25, the weight energy consumption rises by about 15%, which verifies the robustness of the proposed MARL approach.

Fig. 12 shows weighted energy consumption versus the maximum task size $D_{\max}$ and evaluates the without DT scheme ("labeled as w/o DT"), while fixing the minimum value 5×10^5 for with different number of MUs. The w/o DT scheme is evaluated by adding the random deviation rate in [-0.5,0.5] on the normalized variables of locations and computational frequency. This is under the following
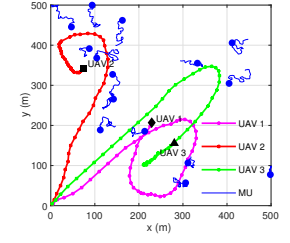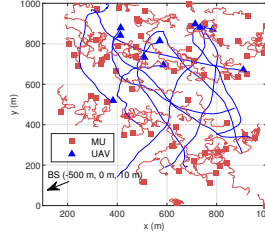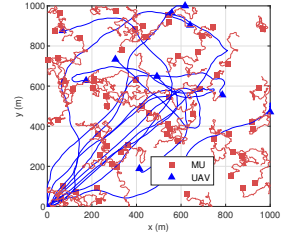
consideration. The DT-assisted framework enables UAVs to directly synchronize the estimated information with the DT layer at BS. Hence, the estimated deviation is cut down by avoiding the frequent queries for exchanging the environment information between UAVs, and thus the devices can obtain more timely status of the system. Therefore, the estimated deviation is set higher to evaluate the w/o DT schemes, whose performance is intuitively worse than that of DT-assisted schemes. Additionally, it can be readily observed that more energy is consumed as the task size grows and the bandwidth decreases. It can be explained by the fact that more computation resource is required to satisfy the latency requirements of MUs, and since the bandwidth decreases, the average communication resource decreases. As such, the transmission expense ascends and the MUs tend to compute more tasks locally, thereby leading to the increasing of energy consumption.

Fig. 13 displays the trajectories of MUs and UAVs in different scenarios. In Fig. 13 (a), the MUs are relatively crowded in a part of region, and the UAVs start from random locations, with the width of region 500 m and $T = 50$ s. It can be observed that the trained UAVs are capable of flying closer to MUs and keep hovering on the crowded MUs to pursue higher transmit rate. In Fig. 13 (b), the UAVs start at the corner of the region, and the MUs are relatively dispersed. At the beginning, the UAVs can quickly fly to the MUs, and then cooperate with each other to serve the associated MUs. Additionally, the UAVs are close to most of associated MUs rather than fully involving the remote ones, which indicates that UAVs can skillfully the long-term reward. In Figs. 13 (c) and (d), we present the trajectory with large-scale simulations with $K = 60$ users and $M = 10$ UAVs. It can be seen that the UAVs can move to the regions with more MUs and cooperatively cover the region of MUs to provide better service

and maintain the fairness. It verifies that in the situation with larger scale, the reward function can guide the UAVs to rapidly access the MUs.

V. CONCLUSION

In this paper, we proposed a multi-UAV-assisted MEC network with air-ground cooperation where DT is applied to enhance the offloading service. We formulated a weighted sum energy consumption minimization problem by jointly optimizing the offloading decision, bandwidth, flying trajectory, communication resource and computation resource. To tackle this challenging problem, we modeled our problem as an MDP where MUs and UAVs act as agents to collectively interact with the environment to receive distinctive observations. Considering the high-dimensional hybrid action space, the MAPPO algorithm with attention mechanism and Beta distribution was leveraged to efficiently obtain the optimal policy. Simulation results verified that the proposed scheme can significantly reduce energy consumption of the network compared with other benchmarks. Additionally, the combination of UAV and BS offloading strategies can take fully advantage of communication and computational resources, as well as adapt to the time-varying network environments.

REFERENCES

[1] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6G internet of things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[2] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11 891–11 915, Aug. 2021.

[3] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[4] B. Ji, Y. Wang, K. Song, C. Li, H. Wen, V. G. Menon, and S. Mumtaz, "A survey of computational intelligence for 6G: Key technologies, applications and trends," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7145–7154, Oct. 2021.

[5] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: A distributed learning framework and applications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 56–62, Apr. 2021.

[6] X. Cao, B. Yang, C. Huang, C. Yuen, Y. Zhang, D. Niyato, and Z. Han, "Converged reconfigurable intelligent surface and mobile edge computing for space information networks," *IEEE Netw.*, vol. 35, no. 4, pp. 42–48, Jul. 2021.

[7] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[8] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "AI driven heterogeneous MEC system with UAV assistance for dynamic environment: Challenges and solutions," *IEEE Netw.*, vol. 35, no. 1, pp. 400–408, Jan. 2021.

[9] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for MISO UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, May 2022.

[10] J. Ji, K. Zhu, and L. Cai, "Trajectory and communication design for cache-enabled UAVs in cellular networks: A deep reinforcement learning approach," *IEEE Trans. Mobile Comput.*, to be published, doi:10.1109/TMC.2022.3181308.

[11] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13 789–13 804, Sep. 2021.

[12] S. Mihai, M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, B. Barn, D. Shetve, R. V. Prasad, H. Venkataraman, R. Trestian, and H. X. Nguyen, "Digital twins: A survey on enabling technologies, challenges, trends and future prospects," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2255–2291, Fourthquarter 2022.

[13] Y. Dai and Y. Zhang, "Adaptive digital twin for vehicular edge computing and networks," *J. Commun. Inf. Networks*, vol. 7, no. 1, pp. 48–59, Mar. 2022.

[14] T. Do-Duy, L. D. Nguyen, T. Q. Duong, S. R. Khosravirad, and H. Claussen, "Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3411–3424, Nov. 2021.

[15] D.-H. Tran, V.-D. Nguyen, S. Chatzinotas, T. X. Vu, and B. Ottersten, "UAV relay-assisted emergency communications in IoT networks: Resource allocation and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1621–1637, Nov. 2022.

[16] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 10, pp. 3536–3550, Oct. 2022.

[17] B. Dai, J. Niu, T. Ren, Z. Hu, and M. Atiquzzaman, "Towards energy-efficient scheduling of UAV and base station hybrid enabled mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 915–930, Jan. 2022.

[18] 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.

[19] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[20] T. Cai, Z. Yang, Y. Chen, W. Chen, Z. Zheng, Y. Yu, and H.-N. Dai, "Cooperative data sensing and computation offloading in UAV-assisted crowdsensing with multi-agent deep reinforcement learning," vol. 9, no. 5, pp. 3197–3211, Sep.-Oct. 2022.

[21] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6949–6960, Mar. 2022.

[22] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12 240–12 251, Oct. 2020.

[23] W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, "Adaptive federated learning and digital twin for industrial internet of things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5605–5614, 2021.

[24] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5098–5107, Jul. 2021.

[25] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022.

[26] T. Do-Duy, D. Van Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 11, no. 4, pp. 806–810, Apr. 2022.

[27] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multi-agent deep reinforcement learning for vehicular edge computing and networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1405–1413, Feb. 2022.

[28] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10 863–10 877, Oct. 2022.

[29] 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.

[30] Z. Yang, S. Bi, and Y.-J. A. Zhang, "Online trajectory and resource optimization for stochastic UAV-enabled MEC system," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5629–5643, Jul. 2022.

[31] Y. Xu, T. Zhang, Y. Liu, D. Yang, L. Xiao, and M. Tao, "UAV-assisted MEC networks with aerial and ground cooperation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7712–7727, Dec. 2021.

[32] C. Zhao, J. Liu, M. Sheng, W. Teng, Y. Zheng, and J. Li, "Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3193–3207, Oct. 2021.

[33] 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.

[34] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021. [Online]. Available: http://arxiv.org/abs/2103.01955

[35] Z. Gao, L. Yang, and Y. Dai, "Large-scale computation offloading using a multi-agent reinforcement learning in heterogeneous multi-access edge computing," *IEEE Trans. Mobile Comput.*, to be published, doi:10.1109/TMC.2022.3141080.

[36] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, "Cooperative internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6807–6821, Nov. 2020.

Zesong Fei (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the Beijing Institute of Technology (BIT) in 2004.

He is currently a Professor with the Research Institute of Communication Technology, BIT, where he is also involved in the design of the next-generation high-speed wireless communication. He has authored or coauthored over 50 articles in IEEE journals. His research interests include wireless communications and multimedia signal processing. He serves as a Lead Guest Editor for *Wireless Communications and Mobile Computing*, and *China Communications* Special Issue on Error Control Coding. He is the Chief Investigator of the National Natural Science Foundation of China. He is a Senior Member of the Chinese Institute of Electronics and the China Institute of Communications.

Wenshuai Liu received the B.S. degree from Liren College of Yanshan University, Qinhuangdao, China, in 2019. He is currently pursuing the M.S. degree with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His current research interests include mobile edge computing and deep reinforcement learning.

Bin Li (Member, IEEE) received the Ph.D. degree in information and communication engineering from Beijing Institute of Technology, Beijing, China, in 2019.

From 2013 to 2014, he was a Research Assistant with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong. From 2017 to 2018, he was a Visiting Student with the Department of Informatics, University of Oslo, Oslo, Norway. In 2019, he joined the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include wireless cooperative networks, unmanned-aerial-vehicle communications, and mobile-edge computing.

Wancheng Xie is currently pursuing the B.S. degree with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His current research interests include mobile edge computing and deep reinforcement learning.

Yueyue Dai (Member, IEEE) is currently an Associate Professor with the Research Center of 6G Mobile Communications, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. Her current research interests include mobile edge computing, Internet of Vehicles, blockchain, and deep reinforcement learning. She is currently the Guest Editors of the *IEEE Network*, *Digital Communications and Networks*, and *Future Generation Computer Systems*. The TPC members of IEEE ICC, GLOBECOM, etc.