Journal of Systems Architecture 170 (2026) 103634

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Multi-agent reinforcement learning for resource allocation in NOMA-enhanced aerial edge computing networks

Longxin Zhang [a], Xiaotong Lu [a], Jing Liu [b,*], Yanfen Zhang [a], Jianguo Chen [c], Buqing Cao [a,d], Keqin Li [e]

[a] School of Computer Science and Artificial Intelligence, Hunan University of Technology, Zhuzhou 412007, China
[b] Department Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China
[c] School of Software Engineering, Sun Yat-Sen University, Zhuhai 519082, China
[d] Key Laboratory of Intelligent Sensing System and Security (Hubei University), Ministry of Education, 430062, China
[e] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Aerial edge computing
Deep reinforcement learning
Non-orthogonal multiple access (NOMA)
Resource allocation
Unmanned aerial vehicles (UAVs)

ABSTRACT

The advantages of unmanned aerial vehicles (UAVs) in terms of maneuverability and line-of-sight communication have made aerial edge computing (AEC) a promising solution for processing computationally intensive tasks. However, the constrained computational resources of UAVs and the complexity of multi-UAV coordination pose significant challenges in designing efficient trajectory optimization and power allocation strategies to enhance user service quality. To address this issue, we construct an AEC architecture assisted by non-orthogonal multiple access (NOMA) and a deep reinforcement learning (DRL) algorithm based on dynamic Gaussian mixture and sharing networks (DRL-DGSN). By leveraging the successive interference cancellation technology of NOMA, DRL-DGSN simultaneously optimizes user association, UAV power allocation, and trajectory design to maximize system throughput. First, DRL-DGSN employs a dynamic user association algorithm based on Gaussian mixture model, achieving capacity-aware uniform clustering through probabilistic modeling combined with cluster capacity constraints, effectively preventing UAV overload. Second, DRL-DGSN utilizes a multi-agent DRL framework with a dueling network architecture and double deep Q-network. By integrating a shared network, agents can efficiently share experiences, enabling simultaneous optimization of multi-UAV cooperative trajectories and power allocation, while reducing Q-value overestimation and enhancing training efficiency. Finally, extensive experiments validate the superiority and effectiveness of DRL-DGSN across various scenarios.

1. Introduction

The swift advancement of Internet of Things and wireless communication technologies has triggered a surge in latency-sensitive and computation-intensive tasks. Cloud servers can provide powerful computing capabilities to effectively deal with computation-intensive tasks. However, these servers are frequently distant from end-users, which leads to high transmission latency and energy consumption. Given this background, mobile edge computing (MEC) [1] emerges as a crucial technology for addressing the transmission bottleneck in cloud centers. This technology extends the computing power from traditional cloud centers to the network edge. With this approach, computing tasks are offloaded by wireless devices to mobile edge servers at the network edge for processing, resulting in an increase in computing power [2]. However, in scenarios such as emergency response and remote areas, the wireless coverage of MEC servers is limited, the communication quality is poor, and even the MEC servers may fail to provide services due to damage, which causes difficulty in ensuring a better service quality. Recently, unmanned aerial vehicles (UAVs) have experienced rapid development due to their maneuverability, flexibility, and ease of operation [3]. Thus, researchers have combined UAVs with MEC technology and designed aerial edge computing (AEC) assisted by UAVs. In the AEC architecture, UAV as the carrier of assisted edge computing can effectively solve the problem of coverage blindness. It can provide highly stable line-of-sight (LOS) communication by virtue of its positional advantage, which improves the efficiency of data transmission. Given the unpredictable mobility of users, static UAV deployment struggles to deliver efficient edge computing services. UAVs are required to precisely plan flight trajectories [4] and efficiently

* Corresponding author.
E-mail addresses: longxinzhang@hut.edu.cn (L. Zhang), luxiaotong@stu.hut.edu.cn (X. Lu), luijing_cs@wust.edu.cn (J. Liu), yanfen.z@foxmail.com (Y. Zhang), chenjg33@mail.sysu.edu.cn (J. Chen), buqingcao@hut.edu.cn (B. Cao), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.sysarc.2025.103634
Received 25 May 2025; Received in revised form 28 September 2025; Accepted 16 November 2025
Available online 17 November 2025

allocate transmit power. Thus, research has focused on optimizing the flight trajectories of each UAV to expand the coverage of target user service areas. Moreover, flight safety and reasonable power allocation are guaranteed to maximize throughput and maintain user fairness. Considering the non-convex characteristics of the multi-variable and multi-constraint problems inherent in multi-UAV cooperation, traditional heuristic algorithms frequently demonstrate inefficiency and converge to local optima. This limitation complicates the rapid design of a safe and efficient cooperation scheme that incorporates power allocation within complex environments.

For the core requirement of throughput optimization under transmit power constraints in AEC, non-orthogonal multiple access (NOMA) demonstrates significant technical advantages. NOMA outperforms traditional multiple-access technologies by enhancing user fairness and spectral efficiency through power domain multiplexing and successive interference cancellation (SIC), resulting in higher throughput [5]. In practical AEC scenarios, sparse and scattered terrestrial users exacerbate the complexity of resource management. Therefore, performing reasonable NOMA clustering for users is needed to effectively manage power allocation and reduce inter-user interference. Most existing studies adopt K-means or K-means++ clustering algorithm [6]. However, the traditional K-means algorithm demonstrates limited capability in delineating non-spherical clusters [7]. Thus, it cannot handle the complex and variable user distribution in AEC scenarios effectively. The user count and single-user resource allocation demonstrate an inverse relationship, attributed to the constraints of UAV communication resources. Existing clustering algorithms are mostly based on partitioning, layering, density, or modeling. These algorithms also lack an explicit constraint mechanism for cluster capacity [8]. This leads to over-partitioning of resources in high-density regions, resulting in the inability to guarantee the transmission rate for individual users. Furthermore, the total throughput will decline further due to increased interference, ultimately impacting the overall communication quality of AEC.

Deep reinforcement learning (DRL) integrates the representational learning capabilities of neural networks with the sequential decision-making benefits of reinforcement learning [9]. It learns through interactions with the environment and can adapt to dynamic changes within that environment. In recent years, several studies have utilized DRL to address joint optimization problems in AEC, including the co-optimizing deployment of UAVs and the allocation of communication resources, and joint decision making in task offloading and UAV trajectory control [10]. The complexity of the AEC environment and the vast state and action spaces introduce significant demands on the convergence and generalization capabilities of the algorithm [11]. Furthermore, in a multi-UAV system, each UAV can be regarded as an independent agent when employing DRL technology. Effectively managing and coordinating interactions among these agents to ensure overall performance and synergistic efficiency has become an urgent research challenge.

To tackle the challenges mentioned above, this study concentrates on resource allocation algorithms within NOMA-assisted AEC systems. Moreover, a DRL algorithm based on dynamic Gaussian mixture and shared network (DRL-DGSN) is proposed. DRL-DGSN employs a novel multi-agent DRL (MADRL) framework to optimize the user association strategy, UAV flight trajectories, and communication resource allocation simultaneously. The framework is also used to maximize the system throughput under the premise of guaranteeing user fairness. The primary contributions of this study are highlighted below:

- We construct a NOMA-assisted AEC network architecture that utilizes UAV clusters as low-altitude mobile base stations to provide cooperative communication services to mobile users. In addition, we implement the SIC technique to effectively reduce interference among cluster users. By formulating a mixed-integer optimization problem that integrates user-UAV association policies, UAV flight trajectory control, and transmit power allocation, we maximize system throughput while ensuring the minimum transmission rate for users.
- We propose a dynamic user association algorithm based on the Gaussian mixture model (GMM) (DUA-GMM) to optimize the user-UAV association strategy. The DUA-GMM achieves accurate clustering of user groups by leveraging the advantages of GMM in modeling complex data distributions. Moreover, the upper limit on the number of service users due to UAV energy consumption constraints is addressed by adopting the uniform clustering algorithm (UCA) to constrain cluster capacity, we effectively tackle the challenge of ensuring users' transmission rates in high-density regions under transmit power constraints.
- We model the problem as a Markov decision process (MDP) and develop the DRL-DGSN algorithm to optimize the flight trajectory control and transmit power allocation strategies of UAVs simultaneously. DRL-DGSN employs a double deep Q-network (DDQN) algorithm based on the dueling network architecture, which effectively reduces the overestimation problem in Q-value estimation. Unlike existing MADRL algorithm, the shared network is used among the agents. This way further promotes the information sharing and collaboration among UAVs. Experimental results show that, compared with other state-of-the-art methods, DRL-DGSN achieves an increase in system throughput ranging from 10.06% to 238.04%.

The remainder of the study is organized as follows. Section 2 reviews the related works. Section 3 introduces the system model and outlines the optimization problem. Section 4 outlines the problem to be addressed and provides a formal description. Section 5 discusses the DUA-GMM and DRL-DGSN algorithms in detail. Section 6 assesses and analyzes the performance of DRL-DGSN. Section 7 offers concluding remarks.

2. Related work

The deep integration of MEC and UAV technologies has caused difficulty in efficient resource allocation across dense and heterogeneous environments in the field of edge intelligence. This section offers a thorough overview of the latest advancements in research addressing this challenge.

In MEC, research has mainly focused on the resource optimization of terrestrial fixed servers, with the core objective of reducing terminal latency and energy consumption through computational offloading. Chen et al. [12] developed a three-tier energy consumption model of cloud–edge-end systems and introduced a particle swarm optimization algorithm that incorporates self-adaptation capabilities, further enhanced by genetic algorithm operators. This model achieves efficient offloading decisions at each layer through layer partitioning operations, which effectively reduces coding dimensions and shortens execution times. Chen et al. [13] proposed a task offloading method in cloud–edge computing that is real-time and aware of task dependencies, utilizing Deep Q Networks (DQN) to facilitate parallel task offloading without predefined priority constraints. Liao et al. [14] created a task offloading algorithm that employs a double-layer DRL model, effectively minimizing latency and energy consumption by simultaneously optimizing task offloading decisions, transmission power, and CPU frequency. Sadiki et al. [15] investigated large-scale multiple-input multiple-output MEC systems within the context of computational offloading. They proposed two DRL algorithms, namely, DQN and proximal policy optimization, to minimize power consumption and offloading latency for mobile devices. Pang et al. [16] designed an ultra-dense heterogeneous network scenario with MECs. Additionally, they proposed a distributed framework for task offloading and wireless resource management, aimed at optimizing the task offloading process, scaling local computing frequencies, assigning subchannels, and regulating transmit power. Saberikia

et al. [17] proposed a fault-tolerant scheduling algorithm that combines mixed-integer linear programming (MILP) with heuristics. This approach aims to optimize energy consumption, fault tolerance, and load balancing to effectively manage dynamic task loads in edge-fog-cloud multi-layer architectures. Wu et al. [18] addressed the deficiencies of traditional edge computing networks that rely on a single access point and the privacy concerns associated with the offloading process by presenting a privacy-preserving strategy grounded in stochastic game theory. They developed a framework for offloading tasks using multiple access points and created a model to evaluate the quality of service. These studies have laid a theoretical foundation for resource allocation. However, the static architecture of ground-based MEC faces challenges in adapting to the multi-machine collaboration demands of dynamic scenarios.

Owing to their high maneuverability and advantages in LOS communication, UAVs have gradually emerged as the core carriers of AEC to overcome the physical limitations of ground-based MEC. Li et al. [19] addressed the joint optimization challenge of energy efficiency and fairness in UAV-assisted MEC networks by proposing an optimization algorithm for joint trajectory planning and computational offloading strategy based on energy-saving, which integrates an optimization-embedding multiagent DRL for autonomous decision making under dynamic demand. Hao et al. [20] explored computing offloading in multi-UAV collaborative MEC systems. They innovatively integrated embedding tables with variational autoencoders in a DRL framework to effectively resolve the co-optimization challenges in hybrid action spaces. Luo et al. [21] considered the difficulty for UAVs to achieve ideal search results during the search process due to the constraints of battery life and computational capability. They designed a framework for collaborative target search by multiple UAVs to optimize computational offloading and trajectory design simultaneously under energy and time constraints. Li et al. [22] developed a maximum clique with weighted graph algorithm to maximize the resource utilization of UAVs by simultaneously optimizing the communication range of user devices along with the latency and energy consumption of computational tasks. Ghosh et al. [23] designed a model utilizing a quantum-inspired gravitational search algorithm for the multi-UAV multi-user binary task offloading problem. This model employs quantum coding for decoding and optimizes this process using a hashing method. It incorporates a penalty mechanism to address the issue of violating decoding agents while ensuring polynomial time execution. Yan et al. [24] innovatively modeled the time interval from information generation to reception as the age of information (AoI) for dynamic scenarios involving UAV-assisted vehicle edge computing. The twin delayed deep deterministic policy gradient method was proposed based on Actor-Critic networks, aiming to optimize AoI, energy consumption, and leasing costs. Despite significant advancements in dynamic resource scheduling for UAV-assisted MEC, challenges such as spectrum resource competition and multi-user interference in dense UAV swarm scenarios continue to restrict further improvements in system performance. Traditional orthogonal multiple access (OMA) technologies can reduce interference through orthogonal resource allocation mechanisms. However, their limited spectrum efficiency fails to meet the demands of high-density user access.

To address the limitations of traditional OMA technologies, NOMA offers significant advantages in complex communication scenarios involving dense UAV deployments by enhancing spectrum efficiency through non-orthogonal resource reuse. Hadi and Ghazizadeh [25] implemented user clustering and resource allocation using a three-stage heuristic algorithm in a NOMA framework. However, the clustering method fails to consider load balancing among UAVs. Umar et al. [26] utilized a NOMA clustering approach based on downlink channel gain and developed a pairing method based on the size of offloaded data in the uplink. They constructed an optimization framework using the Lagrangian function to reduce delays by optimizing power and computational resources. However, this approach only addresses scenarios with a single device in each NOMA cluster, which does not accommodate account for multi-user interference. Lu et al. [27] carried out a secure communication scheme for flying eavesdroppers to tackle information security issues. The authors first derived mathematical expressions to clarify the worst-case security scenario. Then, they applied block coordinate descent and successive convex approximation methods aimed at maximizing average secure computational power while ensuring minimum secure computational requirements for each ground user. Xu et al. [28] used secure computational power as the performance metric and introduced a block coordinate descent algorithm to tackle the task offloading decision-making problem. However, these schemes are limited to single-UAV scenarios and overlook the inter-cluster interference caused by multi-computer coordination and the high-dimensional complexities of system modeling. Dai et al. [29] suggested a resource scheduling strategy for multi-UAV scenarios by combining the fuzzy C-means (FCM) algorithm with the MADRL framework to jointly optimize channel allocation, trajectory control, and transmission power. This framework aims to maximize data rates and enhance communication fairness among UAVs while prioritizing ground base stations. FCM assumes implicit spherical clusters based on Euclidean distance, and the DQN algorithm employed in this framework struggles with Q-value overestimation, which can lead UAV action decisions to deviate from the globally optimal solution. In contrast, the DUA-GMM module integrated with the DRL-DGSN algorithm proposed in this study employs probabilistic distribution modeling to address the spherical assumption, accurately fitting the randomly dispersed characteristics of user distributions. Concurrently, it utilizes the DDQN algorithm based on the dueling network to effectively eliminate Q-value estimation bias and enhance the global optimality of multi-agent collaborative decision-making.

The aforementioned work highlights the potential of NOMA in AEC. However, the challenges of dynamic association and collaborative decision making among multiple agents have not yet been systematically addressed. To this end, the DRL-DGSN algorithm is introduced, which integrates a fusion framework of GMM and DRL to tackle the dynamic association requirements of users and the global optimization problem of multi-agent collaboration in the AEC environment.

3. System model

This study constructs a multi-UAV-assisted AEC system based on NOMA technology to address the problem of ground base station service interruption in disaster environments. As shown in Fig. 1, the system provides communication services for multiple terrestrial mobile ground users in a cooperative manner by deploying several UAVs equipped with MEC servers as low-altitude mobile base stations. In this setup, each UAV operates on the same frequency band and employs SIC technology to effectively reduce signal interference within a multi-user clustering environment. Once the ground users are grouped into clusters using a clustering algorithm, each user cluster selects a suitable UAV for association. Then, the UAV allocates transmit power efficiently to the associated terrestrial ground users. The optimization goals of this system include realizing an association strategy between ground users and UAVs, controlling the 3D trajectory of the UAVs, and allocating transmit power, all with the aim of maximizing the total system throughput. The main symbols used in this study are presented in Table 1.

3.1. Mobile model

In this study, a communication network comprising M users and U UAVs is examined, with $M > U$. The sets of users and UAVs are represented by $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{U} = \{1, 2, \ldots, U\}$, respectively. The time series is segmented into T time slots in this scenario. The motion of UAVs and users on the ground is modeled using the Cartesian coordinate system. At time slot t, the position of each UAV u is denoted

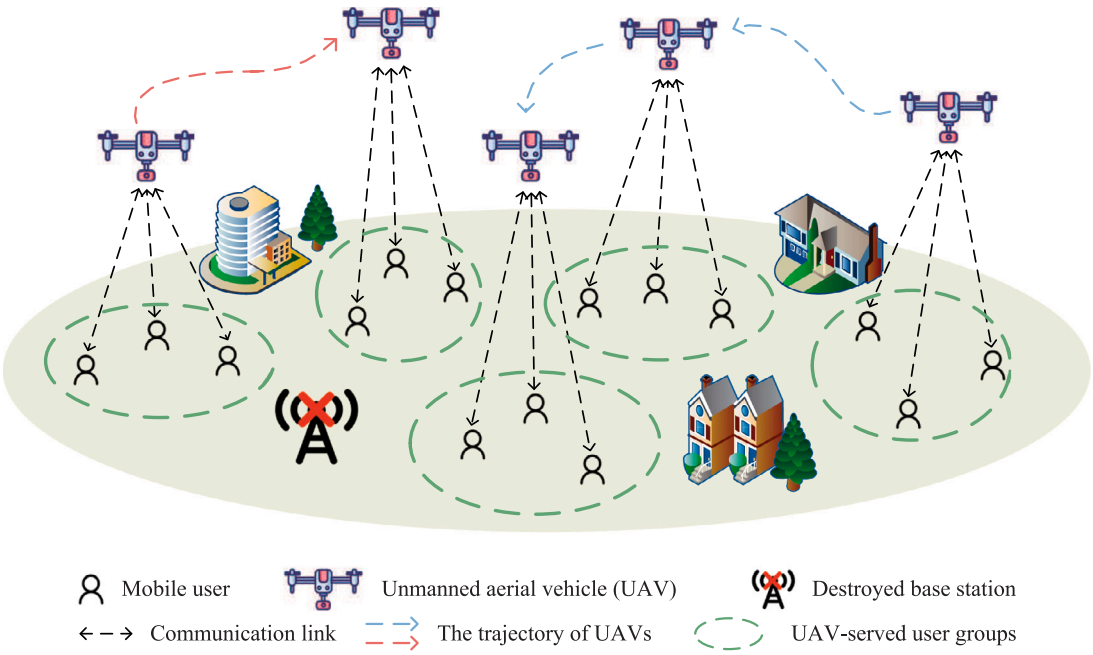

Fig. 1. Multi-UAV assisted AEC system based on NOMA technology.

Table 1
Description of notations.

Notation	Definition
U, M, T	Number of UAVs, users and time slots
$\mathcal{U}, \mathcal{M}, \mathcal{T}$	Sets of UAVs, users and time slots
$d_u(t), d_m(t)$	Coordinates of UAV u and user m at time slot t
$D_{u,m}(t)$	Linear distance between UAV u and user m at time slot t
$d_{\min}$	Minimum distance limit between UAVs
$L_{\text{los}}(t)$, $L_{\text{Nlos}}(t)$	Line-of-sight and non-line-of-sight path loss at time slot t
$L_{\text{avg}}(t)$	Average path loss at time slot t
f_c	Carrier frequency
$R(t)$	Channel fading at time slot t
$g_{u,m}(t)$	Channel gain between UAV u and user m at time slot t
$\mathcal{G}_u(t)$	Set of channel gains between UAV u and its associated users at time slot t
$I^{\text{intra}}_{u,m}(t)$, $I^{\text{inter}}_{u,m}(t)$	Intra-cluster interference and inter-cluster interference experienced by user m
$P_{u,m}(t)$	Transmit power allocated by UAV u to user m at time slot t
N	Noise power
$\varphi_1, \varphi_2, \ldots, \varphi_N$	Decoding sequence of users
$SINR^{\text{sic}}_{u,\varphi_m}(t)$	Signal-to-interference-and-noise ratio of the user φ_m at time slot t
$\tilde{I}^{\text{intra}}_{\varphi_n}(t)$	Intra-cluster interference experienced by user φ_m at time slot t
μ_n	Mean vector of Gaussian components n
Σ_n	Covariance matrix of Gaussian components n
π_n	Mixing coefficients of Gaussian components n
γ_{mn}	Posterior probability between user m and Gaussian components n
$R_{u,\varphi_m}(t)$	Throughput of the user φ_m at time slot t
$\mathcal{R}$	Total system throughput
$K_{\max}$	Maximum number of serviceable users per UAV
K_u	Load of UAV u
o_u	Movement direction of the UAV u

as $d_u(t) = \left(x_u(t), y_u(t), h_u(t)\right)$, where $x_u(t)$, $y_u(t)$ and $h_u(t)$ are the X, Y and Z coordinates of the UAV u. For user m, the position is $d_m(t) = (x_m(t), y_m(t), h_m(t))$, where $x_m(t)$ and $y_m(t)$ are the horizontal coordinates, and the Z coordinate $h_m(t) = 0$ because the user is positioned on the ground.

At time slot t, the linear distance $D_{u,m}(t)$ and the horizontal distance $D^h_{u,m}(t)$ between the UAV u and the user m are denoted as

$$D_{u,m}(t) = \left\| d_u(t) - d_m(t) \right\|_2, \tag{1}$$

$$D^h_{u,m}(t) = \left\| \left(x_u(t) - x_m(t), y_u(t) - y_m(t)\right) \right\|_2. \tag{2}$$

To prevent UAV collisions between UAVs, it must be ensured that the distance between any two UAVs, u and u', is not less than the safe distance, which is expressed by

$$\left\| d_u(t) - d_{u'}(t) \right\|_2 \geq d_{\min}, \forall u, u' \in \mathcal{U}, u \neq u', \tag{3}$$

where $d_{\min}$ is the minimum distance limit between UAVs.

3.2. NOMA-based communication models

During signal transmission, the link transmission loss is influenced by LOS and non-line-of-sight (NLOS) propagation. The LOS path loss

L_{LoS} and the NLOS path loss L_{NLoS} between the UAV u and the user m are modeled as follows [30]:

$$L_{\mathrm{Los}}(t) = 30.9 + \left(22.25 - 0.5 \cdot \lg\left(h_u(t)\right)\right) \cdot \lg\left(D_{u,m}(t)\right) + 20 \cdot \lg\left(f_c\right), \tag{4}$$

$$L_{\mathrm{NLos}}(t) = \max\left\{ L_{\mathrm{Los}}(t), 32.4 + \left(43.2 - 7.6 \cdot \lg\left(h_u(t)\right)\right) \cdot \lg\left(D_{u,m}(t)\right) + 20 \cdot \lg\left(f_c\right) \right\}, \tag{5}$$

where f_c is the carrier frequency.

Let $P_{\mathrm{Los}}(t)$ and $P_{\mathrm{NLos}}(t)$ denote the probabilities of LOS and NLOS propagation, respectively. Their expressions are given as follows:

$$P_{\mathrm{Los}}(t) = \begin{cases} 1, & D^h_{u,m}(t) \le d_c(t); \\ \frac{d_c(t)}{D^h_{u,m}(t)} + \exp\left\{ -\frac{D^h_{u,m}(t)}{P_{loss}(t)} \times \left(1 - \frac{d_c(t)}{D^h_{u,m}(t)}\right) \right\}, & D^h_{u,m}(t) > d_c(t), \end{cases} \tag{6}$$

$$P_{\mathrm{NLos}}(t) = 1 - P_{\mathrm{Los}}(t), \tag{7}$$

where $d_c(t) = \max\left\{294.05 \times \lg\left(h_u(t)\right) - 432.94, 18\right\}$ is the critical distance, and $p_{loss}(t) = 233.98 \times \lg\left(h_{\mathrm{u}}(t)\right) - 0.95$ is the parameter that regulates the attenuation of $P_{\mathrm{Los}}(t)$ [30].

Based on the propagation of LOS and NLOS, the average path loss between the UAV u and the user m is given by [31]

$$\begin{aligned} L_{avg}(t) &= P_{\mathrm{Los}}(t) \cdot L_{\mathrm{Los}}(t) + P_{\mathrm{NLos}}(t) \cdot L_{\mathrm{NLos}}(t) \\ &= P_{\mathrm{Los}}(t) \cdot L_{\mathrm{Los}}(t) + \left(1 - P_{\mathrm{Los}}(t)\right) \cdot L_{\mathrm{NLos}}(t). \end{aligned} \tag{8}$$

The channel gain resulting from random loss $g_{u,m}(t)$ can be evaluated as

$$g_{u,m}(t) = R(t) \times 10^{\left(-\frac{L_{avg}(t)}{10}\right)}, \tag{9}$$

where $R(t)$ represents the channel fading caused by multipath effects [30].

Let $a_{u,m}(t) \in \{0, 1\}$ be a binary variable indicating the association of UAV u with user m. Specifically, $a_{u,m}(t) = 1$ if UAV u is associated with user m at the time slot t, and $a_{u,m}(t) = 0$ otherwise. Given that each user can be associated with a maximum of one UAV at any time slots, the following constraint must be satisfied:

$$\sum_{u=1}^{U} a_{u,m}(t) = 1, \forall m \in \mathcal{M}, t \in \mathcal{T}. \tag{10}$$

The signal power $P^{\mathrm{signal}}_{u,m}(t)$ transmitted by the UAV u to the user m is given by [32]

$$P^{\mathrm{signal}}_{u,m}(t) = a_{u,m}(t) \times P_{u,m}(t) \times g_{u,m}(t), \tag{11}$$

where $P_{u,m}(t)$ represents the power allocated by the UAV u to the user m, satisfying $0 < P_{u,m}(t) < P^{\max}_u$ (with $P^{\max}_u$ denoting the maximum transmit power of UAV u). Meanwhile, the total transmit power allocated by UAV u to its associated users must satisfy $\sum_{m=1}^{M} a_{u,m}(t) P_{u,m} \le P^{\mathrm{total}}_u$, where P^{total}_u is the total transmit power of the UAV u.

The intra-cluster interference $I^{\mathrm{intra}}_{u,m}$ experienced by user m from other users with the UAV u is given by [33]

$$I^{\mathrm{intra}}_{u,m}(t) = \sum_{m_u=1, m_u \neq m}^{M} a_{u,m_u}(t) \times P_{u,m_u}(t) \times g_{u,m_u}(t). \tag{12}$$

The signal interference (inter-cluster interference) $I^{\mathrm{inter}}_{u,m}(t)$ experienced by user m from users served by other UAVs can be expressed as [33]

$$I^{\mathrm{inter}}_{u,m}(t) = \sum_{u_o=1, u_o \neq u}^{U} \sum_{m_o=1}^{M} a_{u_o,m_o}(t) \times P_{u_o,m_o}(t) \times g_{u_o,m_o}(t). \tag{13}$$

The signal-to-interference-and-noise ratio (SINR) for the user m can be calculated by [34]

$$SINR_{u,m}(t) = \frac{P^{\mathrm{signal}}_{u,m}(t)}{I^{\mathrm{intra}}_{u,m}(t) + I^{\mathrm{inter}}_{u,m}(t) + N}, \tag{14}$$

where N is the noise power.

In this system architecture, multiple users transmit their requests to the UAV via the uplink channel, which may result in inter-signal interference due to variations in channel conditions and signal strength between the UAV and the users. We implement the SIC technique to address this challenge. This technique can significantly enhance signal reception quality, optimize spectrum utilization, and adapt to differences in signal strengths of users by progressively decoding stronger signals and mitigating their interference with weaker signals.

Based on a comprehensive consideration of channel gain, interference factors, and noise levels, we adopt SINR as the core metric for determining the decoding order. Specifically, by calculating the SINR, the UAV can accurately assess the strength of the request signal of each user and optimize the processing order accordingly. In this process, received signals are decoded based on SINR priority; that is, user requests with the highest SINR are prioritized, and the interference caused by the decoded requests is gradually mitigated during the decoding process. For the UAV u, when multiple users transmit signals, the SINR ordering can be evaluated by

$$SINR_{u,\varphi_1}(t) \ge SINR_{u,\varphi_2}(t) \ge \cdots \ge SINR_{u,\varphi_{\mathrm{N}}}(t). \tag{15}$$

Therefore, the decoding sequence is $\varphi_1, \varphi_2, \ldots, \varphi_{\mathrm{N}}$. When user φ_1 to user φ_{m-1} have been decoded sequentially, all decoded signals from user $\varphi_1, \varphi_2, \ldots, \varphi_{\mathrm{m-1}}$ are also removed according to the SIC decoding principle. At this point, the intra-cluster interference $\tilde{I}^{\mathrm{intra}}_{\varphi_m}(t)$ for user φ_{m} can be further described as follows:

$$\tilde{I}^{\mathrm{intra}}_{\varphi_m}(t) = \sum_{j=m}^{M} a_{u,\varphi_j}(t) \times P_{\varphi_j}(t) \times g_{u,\varphi_j}(t). \tag{16}$$

The SINR for user φ_m is represented as follows:

$$SINR^{\mathrm{sic}}_{u,\varphi_m}(t) = \frac{P^{\mathrm{signal}}_{u,\varphi_m}(t)}{\tilde{I}^{\mathrm{intra}}_{u,\varphi_m}(t) + I^{\mathrm{inter}}_{u,\varphi_m}(t) + N}. \tag{17}$$

The data offloading transmission throughput $R_{u,\varphi_m}(t)$ for the user φ_m is calculated based on Shannon's theorem [35], and it is formulated as follows:

$$R_{u,\varphi_m}(t) = B \cdot \log_2\left(1 + SINR^{\mathrm{sic}}_{u,\varphi_m}(t)\right), \tag{18}$$

where B is the bandwidth of the UAV u.

4. Problem formulation

The objective of this study, denoted as P_1, is to maximize the total network throughput by jointly optimizing the user association policy, UAV trajectory planning and transmit power allocation, expressed as:

$$\mathrm{P}_1: \quad \max \mathcal{R} = \sum_{t=0}^{T} \sum_{u=1}^{U} \sum_{m=1}^{N} R_{u,\varphi_m}(t),$$

$$\text{s.t. } x_u \in [x_{\min}, x_{\max}], \forall u, \tag{19a}$$

$$y_u \in [y_{\min}, y_{\max}], \forall u, \tag{19b}$$

$$h_u \in [h_{\min}, h_{\max}], \forall u, \tag{19c}$$

$$\left\| d_u(t) - d_{u'}(t) \right\|_2 \ge d_{\min}, \forall u, u', u \neq u', \forall t, \tag{19d}$$

$$a_{u,m}(t) \in \{0, 1\}, \forall u, \forall m, \forall t, \tag{19e}$$

$$\sum_{u=1}^{U} a_{u,m}(t) = 1, \forall u, \forall m, \forall t, \tag{19f}$$

$$\sum_{m=1}^{M} a_{u,m}(t) \times p_{u,m} \le p^{\max}_u, \forall u, \forall m, \forall t, \tag{19g}$$

$$R_{u,m}(t) \ge R_{\min}, \forall u, \forall m, \forall t, \tag{19h}$$

where Eqs. (19a) to (19c) define the physical spatial boundaries for UAV flights. Eq. (19d) addresses the requirement for safe distances between UAVs to prevent airborne collisions. Eqs. (19e) and (19f) ensure that each user can and must associate with only one UAV at any given time slot. Eq. (19g) establishes an upper limit on the maximum transmit power of the UAVs, and Eq. (19h) specifies the minimum throughput requirement per user.

5. Algorithm design

In this section, we first introduce the DUA-GMM algorithm for user-UAV association. Next, we present the DRL-DGSN joint optimization approach for UAV trajectory and transmit power allocation. Finally, we analyze the complexity of the proposed algorithms.

5.1. The DUA-GMM algorithm

In AEC scenarios, the dynamic changes in user mobility and distribution introduce challenges for the association between UAVs and users. In this section, the DUA-GMM algorithm is proposed and combined with the UCA algorithm to achieve dynamic association between UAVs and users while ensuring UAV load balancing.

In this study, GMM is utilized to cluster the user population. Notably, each UAV corresponds to a specific cluster. In this mechanism, user requests within the current cluster are offloaded to the assigned UAV for processing, which ensures that the number of clusters matches the number of UAVs.

Specifically, we first randomly initialize the mean vector μ_n and covariance matrix Σ_n, with an equal number of Gaussian components as UAVs. In addition, we assign mixing coefficients π_n for each Gaussian component, with verification that they sum to 1, as described below:

$$\sum_{n=1}^{U} \pi_n = 1. \tag{20}$$

Subsequently, the expectation–maximization (EM) algorithm is iterated until the parameters converge. The EM algorithm comprises two primary phases: the expectation step (E-step) and the maximization step (M-step).

5.1.1. E-step

In the E-step of the EM algorithm, the posterior probability γ_{mn} (i.e., the degree of responsibility) that each user m belongs to each Gaussian component n is computed, which is expressed as

$$\gamma_{mn} = \frac{\pi_n \cdot \mathcal{N}\left(d_m \mid \mu_n, \Sigma_n\right)}{\sum_{j=1}^{U} \pi_j \cdot \mathcal{N}\left(d_m \mid \mu_j, \Sigma_j\right)}, \tag{21}$$

where $\mathcal{N}\left(d_m \mid \mu_n, \Sigma_n\right)$ is the probability density function that follows a Gaussian distribution, with μ_n representing the mean and Σ_n denoting the covariance. $\gamma_{mn} \in [0, 1]$ represents the conditional probability of user m belonging to cluster n. Unlike the hard assignment of the K-means algorithm (i.e., each user can only belong to a single cluster), Eq. (21) adopts a soft assignment mechanism. This mechanism allows each user to be associated to multiple clusters with different probabilities, thereby quantifying the strength of user-cluster associations.

5.1.2. M-step

The responsibility γ_{mn} is used to update the critical parameters of the Gaussian component: the mean vector μ_n iteratively refines the center of Gaussian component n by computing a weighted average of all users' positions. The covariance matrix σ_n characterizes the dispersion and spatial distribution of users around their centers μ_n. The mixing coefficients π_n indicate the expected proportion of users associated with Gaussian component n by averaging the responsibility γ_{mn} of each user. The update formulas are expressed as follows:

$$\mu_n = \frac{\sum_{m=1}^{M} \gamma_{mn} d_m}{\sum_{m=1}^{M} \gamma_{mn}} \tag{22}$$

$$\Sigma_n = \frac{\sum_{m=1}^{M} \gamma_{mn} \left(d_m - \mu_n\right)\left(d_m - \mu_n\right)^T}{\sum_{m=1}^{M} \gamma_{mn}}, \tag{23}$$

$$\pi_n = \frac{\sum_{m=1}^{M} \gamma_{mn}}{M}. \tag{24}$$

The abovementioned steps are iteratively repeated until the parameters converge, which leads to the final assignments of each user to their respective clusters.

UAVs are constrained by limited battery capacity and power when performing tasks. Thus, we address the potential for a sudden drop in endurance due to overloading by setting an upper limit on the maximum number of serviceable users $K_{\max}$ or each UAV. This setting satisfies the condition $K_{\max} = \frac{M}{U}$, which ensures fairness in resource allocation among clusters. The load K_u of each UAV u should converge to $K_{\max}$ to maximize the satisfaction rate of user requests. However, traditional GMM clustering approaches only assign users to the nearest cluster centers, ignoring the capacity limitations of those clusters. This approach may result in some UAVs becoming overloaded while others remain idle. To address this issue, we propose the UCA. After users complete GMM clustering, the UCA achieves load balancing between clusters. The UCA iteratively selects an unsaturated cluster n (i.e., $K_n < K_{\max}$), calculates the distance between its center μ_n and all users exceeding the capacity limit, and reassigns the nearest users to cluster n. This process repeats until all clusters meet the capacity constraint. Ultimately, each balanced cluster is assigned to a UAV, ensuring that the UAV's load equals $K_{\max}$.

The UCA algorithm is designed to overcome this limitation, and its implementation steps are detailed in Algorithm 1. $u2c[\,]$ represents the list of user-cluster associations, $u2c[m]$ indicates the cluster associated with user m, and the mean vector of Gaussian components serves as the cluster center. First, the number of users in each cluster is calculated (lines 1–5). Second, for each cluster n that has not yet reached the capacity limit $K_{\max}$, the distance of all users in the large-scale clusters from the center μ_n of cluster n are calculated (lines 6–13). Finally, the user $m_{\min}$ closest to μ_n is selected and re-associate to cluster n, followed by an update of the cluster size (lines 14–19).

Although the UCA employs a greedy algorithm for local adjustments to address load imbalance, which may lead to local optima, it effectively mitigates the combinatorial explosion issue associated with global optimization. Global optimization requires traversing all user-cluster assignment combinations that satisfy the $K_{\max}$ constraint, resulting in a computational complexity of $O\left(U^M\right)$. This significantly increases decision latency for UAVs, which fails to meet user demands for low-latency responses. In contrast, UCA reduces complexity to $O(K_{\max} \cdot M \cdot U)$ by locally redistributing overloaded users. Furthermore, UCA prioritizes users that are closest to the target cluster center during redistribution, thereby maximizing the preservation of spatial probability distribution information within GMM.

Given that the mobility characteristics of users are modeled as random wandering, the initially established cluster structure becomes less effective over time when traditional one-shot clustering methods are applied. To overcome this challenge, we recalculate the cluster structure by activating the DUA-GMM algorithm at the start of each time slot by using the newly obtained user location data. This method effectively reduces the risk of cluster structure failure due to mobility by periodically clustering and associating users. Algorithm 2 presents the pseudo-code for the DUA-GMM algorithm. Here, $c2u[\,]$ and $u2u[\,]$ denote the association lists of users with clusters and clusters with UAVs, respectively; $c2u[n]$ represents the UAV associated with cluster n; and $u2u[m]$ denotes the UAV associated with user m.

Algorithm 1 UCA

Input: user location d_m, maximum cluster capacity $K_{\max}$, cluster center μ_n, GMM's clustering result $u2c[\ ]$.
Output: optimized user uniform clustering decision $u2c[\ ]$.

1: **for** $m \leftarrow 1$ to M **do**
2: $n\_orig \leftarrow u2c[m]$; // Obtain the original cluster n_orig associated with user m;
3: $K_{n\_orig} \leftarrow K_{n\_orig} + 1$;
4: **end for**
5: **for** $n \leftarrow 1$ to U **do**
6: **while** $K_n < K_{\max}$ **do**
7: **for** $m \leftarrow 1$ to M **do**
8: $n' \leftarrow u2c[m]$;
9: **if** $K_{n'} > K_{\max}$ **then**
10: $D_{n,m} \leftarrow \left\|\mu_n - d_m\right\|_2$;
11: **end if**
12: **end for**
13: Obtain the cluster $\tilde{n}$ associated with $m_{\min}$, where $\tilde{n} = u2c[m_{\min}]$;
14: $u2c[m_{\min}] \leftarrow n$; // Update the cluster associated with user $m_{\min}$ to n;
15: $K_n \leftarrow K_n + 1$, $K_{\tilde{n}} \leftarrow K_{\tilde{n}} + 1$;
16: **end while**
17: **end for**

Algorithm 2 DUA-GMM

Input: UAV location d_u, user location d_m, maximum cluster capacity $K_{\max}$.
Output: UAV and user association decisions $u2u[\ ]$.

1: Initialize the mixing coefficients π_n by using Eq. (20);
2: Initialize the mean vector μ_n and the covariance matrix Σ_n;
3: **for** $i \leftarrow 1$ to $i_{\max}$ **do**
4: Initialize the array $u2c[\ ]$;
5: **for** $m \leftarrow 1$ to M **do**
6: **for** $n \leftarrow 1$ to U **do**
7: Calculate γ_{mn} by using Eq. (21);
8: **end for**
9: $u2c[m] \leftarrow \arg\max_n r_{nm}$;
10: **end for**
11: **for** $n \leftarrow 1$ to U **do**
12: Calculate μ_n and Σ_n by using Eqs. (22) and (23);
13: Update π_n by using Eq. (24);
14: **end for**
15: **end for**
16: Call Algorithm 1;
17: **for** $n \leftarrow 1$ to U **do**
18: $c2u[n] \leftarrow \arg\max_u \left\|d_u - \mu_n\right\|$;
19: **end for**
20: **for** $m \leftarrow 1$ to M **do**
21: $n' \leftarrow u2c[m]$ // Obtain the cluster n' associated with user m;
22: $u2u[m] \leftarrow c2u\left[n'\right]$;
23: **end for**

The DUA-GMM algorithm starts by initializing the mixing coefficients π_n, the mean vector μ_n, and the covariance matrix Σ_n (lines 1–2). Next, the degree of responsibility γ_{mn} is calculated between each user m and each Gaussian component n, with user m being associated with the Gaussian component that has the highest degree of responsibility (lines 3–10). Following this, the algorithm iterates through all Gaussian components to update the mean vector μ_n, the covariance matrix Σ_n and the mixing coefficients π_n, refining the Gaussian model to better capture the distribution of the user (lines 11–15). After completing the GMM clustering, Algorithm 1 is executed to uniformly cluster the users (line 16). Finally, each UAV is assigned to the nearest cluster in succession, ensuring that every UAV is matched with the closest cluster, thereby establishing the association between UAVs and users (lines 17–23).

5.2. Resource allocation algorithm based on DRL-DGSN

It is widely acknowledged that the mixed-integer nonlinear programming problem examined is NP-hard [36]. Consequently, traditional convex optimization methods encounter great challenges when addressing such complex problems. To solve the problem P_1, it is first formulated as a MDP, and the DRL-DGSN algorithm is developed to simultaneously optimize the power allocation and trajectory control strategies for the UAV.

5.2.1. MDP formulation

In this section, the resource allocation process is modeled within an MDP framework to optimize decision making for UAV agents in communication tasks. During interactions with the environment, the current state is perceived by UAV agents, actions are selected based on their decision-making strategies, and the environmental feedback (i.e., reward) obtained after executing an action is utilized as the basis for learning. This iterative process allows for the continuous refinement of their decision-making strategies. Three key components of the MDP–state s, action a, and reward r–are defined as follows.

(1) **State:** The UAV agent can sense the following information: the position of the current UAV relative to other UAVs, as well as the channel gains between the current UAV and other UAVs in conjunction with their associated users. Based on this information, the state of the UAV agent is characterized as

$$s = \left\{d_u(t), \{d_v(t)\}_{v\in\mathcal{U}\setminus\{u\}}, \mathcal{G}_u(t), \left\{\mathcal{G}_v(t)\right\}_{v\in\mathcal{U}\setminus\{u\}}\right\}, \tag{25}$$

where $\mathcal{U} \setminus \{u\}$ denotes the set of UAVs $\mathcal{U}$ except UAV u. $\mathcal{G}_u(t) = \left\{g_{u,m}\right\}_{m\in\mathcal{M}}$ represents the set of channel gains between UAV u and its associated users.

(2) **Action:** The UAV must determine its current direction of movement and the transmit power assignment for the associated user within each time slot. The transmit power is selected from a predefined set of discrete levels $\mathcal{P} = \{P_1, P_2, \ldots, P_L\}$, where L is the total number of discrete transmit power levels. Therefore, the action of the UAV agent is described by

$$a = \left\{o_u, P_{u,1}, \ldots, P_{u,K_{\max}}\right\}, \forall u \in \mathcal{U}, \tag{26}$$

where o_u indicates the direction of movement of the UAV u.

(3) **Reward:** The reward function is designed as a combination of system throughput and a penalty mechanism. When the transmission rate of the worst-performing user does not meet the minimum requirement, the system imposes a fixed penalty value η to ensure dual optimization of system performance and fairness. The reward function for the UAV is defined as follows:

$$r = \begin{cases} -\eta, & \text{if the minimum rate is below the threshold;} \\ \sum_{m=1}^{K_{\max}} R_{u,m}(t), & \text{otherwise.} \end{cases} \tag{27}$$

5.2.2. Network architecture of DRL-DGSN

DRL-DGSN is based on the DQN framework for algorithmic improvement. It combines the overestimation suppression mechanism of DDQN with the state–action decoupling idea of dueling network. The Q-value update process of the overall network architecture can be formally described as follows:

(1) **Basic DQN framework**

The DQN algorithm is classified as an off-policy method and is categorized as a value-based reinforcement learning algorithm. Its primary concept involves parameterizing the Q-value function through the construction of a neural network. The iterative update process for the target Q-value $Q(s,a;\theta)$ can be formulated as follows:

$$Q(s,a;\theta) = r + \gamma \max_{a'} Q\left(s',a';\theta'\right), \tag{28}$$

where θ and θ' represent the weights of the current network and the target network, respectively; s' and a' denote the next state and the next action, respectively; and γ signifies the discount factor.

The loss function $Loss$ is defined by

$$Loss = \left(Q(s,a;\theta) - \left(r + \gamma \max_{a'} Q\left(s',a';\theta'\right)\right)\right)^2. \tag{29}$$

The DQN algorithm incorporates an experience replay buffer and a target network to enhance the accuracy of Q-value estimation. The agent acquires experience (s,a,r,s') through interactions with the environment and stores them in the experience replay buffer. Subsequently, random sampling is employed to reduce the impact of temporal correlations on the learning process. The target network is designed to compute the target Q-values and is updated periodically to stabilize the Q-value update process.

(2) **Improvement of DDQN**

In DQN, employing the same network for both action selection and Q-value evaluation introduces maximization bias, which causes the Q-values to be overestimated. To tackle this issue, the DDQN algorithm, which shares a similar structure with DQN, incorporates two separate Q-functions. Specifically, in DDQN, the main network determines the best action for the next state, while the Q-value of that action is computed using the target network. This mechanism, which decouples action selection from Q-value evaluation, effectively mitigates the Q-value overestimation. The update of its Q-value $Q(s,a;\theta)$ is assessed by

$$Q(s,a;\theta) = r + \gamma Q\left(s', \arg\max_{a'} Q\left(s',a';\theta\right);\theta'\right). \tag{30}$$

(3) **Dueling network architecture**

To facilitate independent modeling of state values and action advantages, the DRL-DGSN algorithm employs the dueling network architecture to systematically decompose the Q-value $Q(s,a;\theta)$ into two distinct components: a state value function $V\left(s;\theta_V\right)$ and an advantage function $A\left(s,a;\theta_A\right)$. The state value function $V\left(s;\theta_V\right)$ is employed to evaluate the long-term expected return of the state s. In contrast, the advantage function $A\left(s,a;\theta_A\right)$ measures how much action a exceeds the average value of all possible actions in the same state. This separation facilitates the ability of the network to more effectively learn the intrinsic value of states and the differences in value among various actions. This design feature significantly improves the efficiency of the learning process and enhances its generalization capability. The final output of the Q-network is represented as a linear combination of the two components, as follows:

$$Q(s,a;\theta) = V\left(s;\theta_V\right) + A\left(s,a;\theta_A\right). \tag{31}$$

To enhance the comparability of the output from $A\left(s,a;\theta_A\right)$, zero-averaging is applied to the advantage values. This approach ensures that the advantage values of different actions are assessed on a uniform scale, which facilitates highly effective comparisons, as follows:

$$A\left(s,a;\theta_A\right) \leftarrow A\left(s,a;\theta_A\right) - \frac{1}{|\mathcal{A}|}\sum_{a'\in\mathcal{A}} A\left(s,a';\theta_A\right), \tag{32}$$

where $\mathcal{A}$ is the set of actions.

5.2.3. Design of the DRL-DGSN

In the DRL-DGSN algorithm, each UAV is regarded as an independent agent connected to a shared neural network through state abstraction. This framework facilitates the sharing and reorganization of experiences to train a common neural network. During the training process, data exchange among the agents is implemented to enhance overall performance. Notably, no additional communication between UAVs is required during operation, which further reduces computational complexity. Although each UAV independently selects its actions, this cooperative principle is based on the consideration of other UAV positions as integral components of the environment. This approach allows the UAVs to maintain appropriate distances and avoid interference with one another. Accordingly, effectively collaboration on tasks is facilitated to improve efficiency and performance. Pseudo-code and framework diagrams of the DRL-DGSN algorithm are presented in Algorithm 3 and Fig. 2, respectively.

Algorithm 3 DRL-DGSN

Input: UAV position d_u, user position d_m, channel gain of user g_m.
Output: UAV trajectory design and transmit power allocation.

1: Initialize the experience replay pool D;
2: Initialize $Q(s,a;\theta)$, θ, $Q\left(s',a';\theta'\right)$, $\theta' \leftarrow \theta$;
3: **for** episode $\leftarrow$ 1 to E **do**
4: Initialize state s;
5: **for** $t \leftarrow$ 1 to T **do**
6: Invoke Algorithm 2 at every Δ times slots;
7: **for** $u \leftarrow$ 1 to U **do**
8: Choose an action a based on $\varepsilon-$greedy policy.
9: Execute the action a, Calculate the reward r and obtain the next state s';
10: Store the experience $\left(s,a,r,s'\right)$ into D;
11: Sample random minibatch of experiences from D;
12: **for** each experience $\left(s_j,a_j,r_j,s'_j\right)$ **do**
13: Calculate $V\left(s_j;\theta\right)$ and $A\left(s_j,a_j;\theta\right)$;
14: Calculate $Q\left(s_j,a_j;\theta\right)$ by using $Q\left(s_j,a_j;\theta\right) = V\left(s_j;\theta\right) + \left(A\left(s_j,a_j;\theta\right) - \frac{1}{|\mathcal{A}|}\sum_{a'_j\in\mathcal{A}} A\left(s_j,a'_j;\theta\right)\right)$;
15: Calculate the target Q-value Q_{target} by using $Q_{\text{target}} = r_j + \gamma \cdot Q\left(s'_j, \arg\max_a Q\left(s'_j,a_j;\theta'\right)\right)$;
16: Calculate $Loss$ by using $Loss = \left(Q\left(s_j,a_j;\theta\right) - Q_{\text{target}}\right)^2$;
17: **end for**
18: Perform backpropagation and update θ;
19: Periodically Copy $\theta' \leftarrow \theta$;
20: **end for**
21: **end for**
22: **end for**

In the initialization phase (lines 1–3), the experience replay pool is created to store agent experiences gathered from environmental interactions. Concurrently, the main network is initialized with random weights to encourage exploration and avoid premature convergence to local optima. The target network is also initialized with parameters identical to those of the main network to ensure training stability; it will later be used to compute the target Q-values. During the environment interaction and experience storage phase (lines 4–11), Algorithm 1 is initially invoked for user association at each time step. Subsequently, each UAV agent adopts the $\varepsilon-$greedy strategy to select an action a in the current environment. This interaction yields a reward r and the next state s'. The agent subsequently stores the experience (s,a,r,s') in the experience replay pool.

The parameter computation and update phase (lines 12–24) begins after environment interactions and experience storage are completed. In this phase, the DRL-DGSN algorithm performs stochastic sampling on the experience replay buffer to extract a mini-batch of experiences. For each extracted experience (s,a,r,s'), $V\left(s;\theta_V\right)$, $A\left(s,a;\theta_A\right)$, the Q-value and loss for each network, are computed sequentially. Backpropagation is performed to update θ. In addition, θ' are updated at regular intervals to enhance training stability.

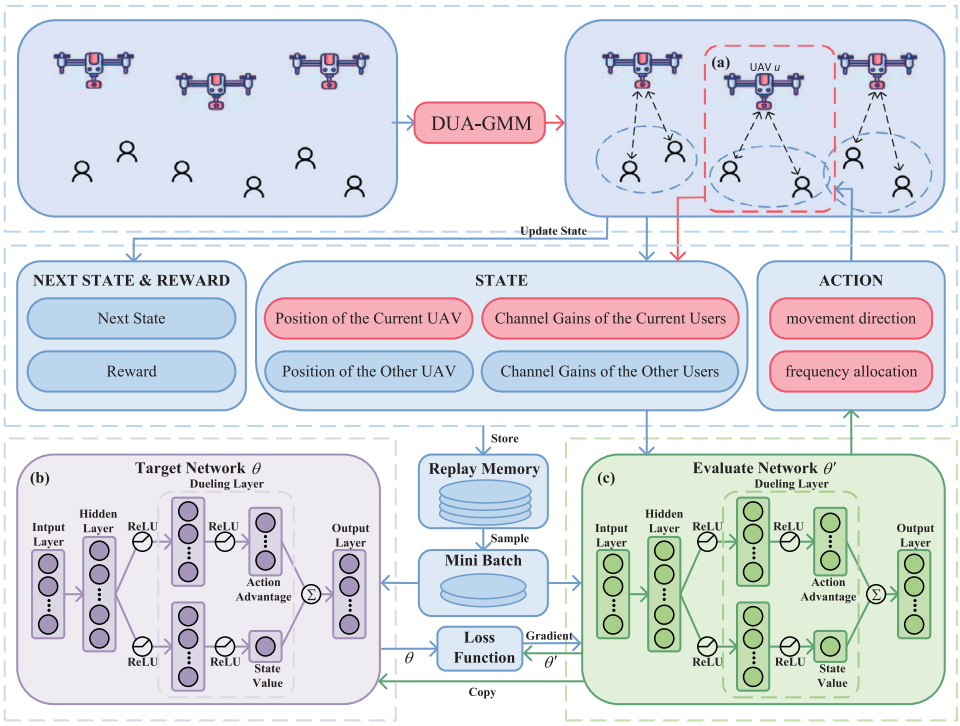

Fig. 2. Framework of DRL-DGSN. (a) represents a single training interaction unit, which includes the current UAV and its associated users, where DRL-DGSN trains each UAV agent individually based on this unit. (b) and (c) together form the shared network module, allowing UAVs to exchange position and channel state information to facilitate the dynamic optimization of collaborative decision-making.

5.2.4. Complexity analysis

The time complexity of the UCA algorithm is denoted as $O(M + K_{\max} \cdot M \cdot U) = O\left(M + \frac{M}{U} \cdot M \cdot U\right) = O(M^2)$. In the DUA-GMM algorithm, the time complexity associated with the GMM clustering based on the 2D coordinates of users is denoted as $O(T_{gmm} \cdot M \cdot U)$, where T_{gmm} represents the total number of iterations for the GMM. When UAVs are associated with clusters, the complexity is $O(M + U)$, resulting in an overall time complexity of the DUA-GMM algorithm of $O(T_{gmm} \cdot M \cdot U + M^2 + M + U) = O(T_{gmm} \cdot M \cdot U)$.

The complexity of the DRL-DGSN algorithm is mainly determined by the complexity of the DUA-GMM algorithm and the complexity associated with the DRL module. The computational complexity of the DRL module is significantly affected by the architectural parameters of the neural network. Let n_1 denote the number of nodes in the shared feature layer, and n_2 denote the number of nodes in the hidden layers of both the value stream and the advantage stream. The computational complexity for a single training iteration of the neural network is represented as $O((|S| \cdot n_1 + 2n_1 \cdot n_2 + n_2 \cdot 1 + n_2 \cdot |\mathcal{A}|) \cdot B)$ [37], which is further simplified to $O((|S| \cdot n_1 + n_2 \cdot |\mathcal{A}|) \cdot B)$, where B denotes the batch size and S is the set of states.

By integrating from the DUA-GMM algorithm and the model training process, the total complexity for a single step of training in the DRL-DGSN algorithm is $O(T_{\text{gmm}} \cdot M \cdot U + (|S| \cdot n_1 + n_2 \cdot |\mathcal{A}|) \cdot B)$. Finally, the overall time complexity of the DRL-DGSN algorithm can be expressed as $O((T_{\text{gmm}} \cdot M \cdot U + (|S| \cdot n_1 + n_2 \cdot |\mathcal{A}|) \cdot B) \cdot T \cdot E)$, where T represents the number of decision steps in a single iteration, and E indicates the total number of iterations throughout the execution of the DRL-DGSN algorithm.

In the DRL-DGSN algorithm, the state space encompasses the 3D spatial coordinates of the UAVs along with the channel gain of the users, and the action space consists of the seven movement directions of the UAVs and three levels of power allocation. Consequently, the state space dimension $|S| = 3U + M$ and the action space dimension $|\mathcal{A}| = 7U + 3M$ are defined. Thus, the time complexity of the DRL-DGSN algorithm is $O((T_{\text{gmm}} \cdot M \cdot U + ((3U + M) \cdot n_1 + (7U + 3M) \cdot n_2) \cdot B) \cdot T \cdot E)$.

Next, we analyze the space complexity of DRL-DGSN. In DRL-DGSN, we employ an experience replay pool to store the interaction experiences between the UAV agent and the environment. These experiences support the online updates of the evaluate network and the target network, and their storage scale directly impacts the space complexity of DRL-DGSN. In DRL-DGSN, the reward dimension is $|\mathcal{R}| = 1$. Therefore, the spatial scale of a single interaction experience (s, a, r, s') is $O(|S| + |\mathcal{A}| + 1 + |S|) = O(2|S| + |\mathcal{A}|)$. Let the experience replay pool capacity be C, then the spatial complexity of DRL-DGSN is $O(C \cdot (2|S| + |\mathcal{A}|)) = O(C \cdot (7U + 5M))$.

6. Performance evaluation

A series of experiments is conducted to validate the performance of the proposed algorithm. Initially, the configurations of the experimental environment and the corresponding parameter settings are detailed. Subsequently, the convergence of the DRL-DGSN algorithm is analyzed. Finally, we carry out a comparative assessment of DRL-DGSN against existing state-of-the-art schemes.

6.1. Experimental setup

In this study, the experiments are designed to establish an AEC network consisting of $U = 3$ UAVs and $M = 6$ ground mobile users within a square area with a dimension of 500 m × 500 m. The initial positions of the UAVs are uniformly distributed throughout this square area at an altitude of 100 m, and their flight altitudes are constrained within the range of [20, 120] m. Ground users are randomly positioned within three designated hotspot areas [38], and each user follows a random roaming model with a speed limit ranging from [0, 0.5] m/s. The deep learning framework adopts a dueling network architecture. It is divided into three modules: the shared feature layer, the state value

Table 2
List of experimental parameters [19,30].

Parameter description	Value
Service area boundaries (x_{max}, y_{max})	500 m
UAV flight altitude range (h_{min}, h_{max})	[20, 120] m
Number of UAVs (U)	3
Number of users (M)	6
Maximum user mobility speed	0.5 m/s
UAV mobility speed	5 m/s
Minimum distance limit between UAVs (d_{min})	10 m
Carrier frequency (f_c)	2 GHz
Bandwidth (B)	30 kHz
Noise power (N)	−60 dBm
Channel fading ($R(t)$)	Rayleigh fading
Maximum UAV transmit power (p^{max})	28.45 dBm
Number of training episodes (E)	200
Number of time slots (T)	60
Update interval of DUA-GMM (Δ)	4
Initial value of ε-greedy exploration	0.9
Learning rate	0.00001
Discount factor (γ)	0.999
Batch size	64
Replay buffer size (C)	50 000

stream, and the advantage stream. All layers utilize ReLU activation functions. The shared feature layer consists of a fully connected layer with 128 nodes. Each hidden layer of the state value stream and the advantage stream contains a fully connected layer with 64 nodes.

During the training phase, the experience replay cache is set with a capacity of 10,000 records. During each model update, 64 experiences are randomly sampled to create a mini-batch. The learning rate is set to 0.00001, and the Adam optimizer is employed for updating gradients. Furthermore, a discount factor $\gamma = 0.999$ is utilized to balance immediate rewards with long-term gains. Table 2 presents the basic parameter settings.

The following four algorithms are selected for comparison to comprehensively evaluate the performance of the proposed DRL-DGSN algorithm.

- Mutual DQN (MDQN) [30]: This algorithm dynamically adjusts UAV deployment locations through periodic K-means clustering of mobile users. It subsequently employs MDQN to enhance the trajectory planning and power allocation of UAVs.
- UAV path selection and resource offloading algorithm (UPRA) [39]: This algorithm combines the K-means clustering method with the Hungarian algorithm to enable periodic online matching between users and UAVs. In addition, a semi-fixed hierarchical power control strategy is developed to improve the autonomous obstacle avoidance and power management abilities of the UAV by employing the DQN algorithm.
- DDQN: The algorithm is based on the original DDQN algorithm [40] for the trajectory optimization and power allocation decisions of UAVs. Meanwhile, it incorporates the DUA-GMM and sharing network proposed in this study to optimize user association and multi-agent interaction.
- Dueling DQN: This algorithm utilizes DUA-GMM developed in this work to enable association between UAVs and users. It employs our proposed shared network among multiple agents for training. The Dueling DQN algorithm [41] is applied to optimize the trajectory control and power allocation strategies for UAVs.
- Random: In this algorithm, the DUA-GMM proposed in this study is used to optimize user association decisions, while the UAV randomly selects flight directions and transmit power.

6.2. Analysis and setting of hyperparameters

The configuration of hyperparameters in deep neural networks plays a crucial role in determining the training efficiency, generalization ability, and decision accuracy of the model. The effects of key hyperparameters, including learning rate, batch size, and discount factor, on the convergence performance of the DRL-DGSN algorithm are systematically analyzed.

First, the convergence of the DRL-DGSN algorithm is investigated using various learning rates, which directly affect the magnitude of gradient updates in the neural network. Fig. 3 depicts the convergence characteristics of the algorithm and highlights significant differences as the learning rate varies from 0.000001 to 0.001. The experimental results indicate that, when the learning rate is set at 0.00001, the DRL-DGSN shows favorable convergence and achieves the maximum return value. By contrast, an excessively high learning rate (e.g., 0.0001) leads to degraded convergence performance, with the convergence curve oscillating around the solution space and failing to reach the optimal solution. Meanwhile, an excessively low learning rate (e.g., 0.000001) results in slow training and convergence to a local optimum. As a result, a learning rate of 0.00001 is chosen for the experiments, as shown in Fig. 3, as it optimally balances convergence speed and model stability.

The batch size, which refers to the number of samples used during neural network training, influences the effect of sample diversity on gradient estimation within the empirical replay mechanism. Three batch sizes are assessed: 32, 64, and 128. As illustrated in Fig. 4, smaller batch sizes (e.g., 32) exhibit significant fluctuations due to increased gradient variations between samples. On the contrary, larger batch sizes (e.g., 128) tend to rely on stale data during training, which potentially hinders convergence. Conversely, a batch size of 64 minimizes gradient variations while preserving sample diversity, which results in the maximum reward value. Therefore, a batch size of 64 is chosen because it optimally accommodates both gradient stability and sample diversity requirements, as evidenced by the results in Fig. 4.

The discount factor γ governs the trade-off between immediate and future rewards. We set the range of γ between 0.9 and 0.9999 [41,42]. As illustrated in Fig. 5, an excessively large discount factor (e.g., 0.9999) leads the model to disproportionately prioritize future rewards, which causes strategy updates to be overly influenced by uncertain future states. Conversely, an excessively small discount factor (e.g., 0.9 or 0.99) overemphasizes immediate rewards, which hinders the learning of long-term dependencies. By setting the discount factor to 0.999, DRL-DGSN effectively balances short-term gains with long-term objectives. This workable balance enables rapid convergence of the reward function toward the optimal solution. Consequently, the selection of a discount factor is based on its optimal trade-off between short-term gains and long-term objectives, as shown in Fig. 5.

6.3. Convergence analysis

As illustrated in Fig. 6, all five algorithms demonstrate improved convergence. Notably, DRL-DGSN not only exhibits enhanced stability but also rapidly achieves and maintains the optimal convergence value after 150 iterations. The optimal solution obtained by DRL-DGSN shows improvements of 9.22% and 129.60% compared with those of UPRA and MDQN, respectively. This enhancement can be attributed to the probabilistic clustering model employed by the DUA-GMM algorithm integrated within DRL-DGSN, which is better suited to accommodate the complexities of user distribution. Furthermore, the cluster capacity constraint facilitates load balancing among UAVs and enhances the fairness of user services.

In contrast to DDQN and Dueling DQN that utilize the DUA-GMM algorithm, DRL-DGSN achieves an average improvement of 39.48% in the final convergence value. This advancement is primarily due to the incorporation of a dueling network architecture, which builds upon the DDQN framework. Thus, it enables more efficient state modeling and mitigates the interference from redundant action updates by decoupling the state value from the action advantage. Moreover, DDQN employs two distinct networks for action selection and value evaluation. Thus,

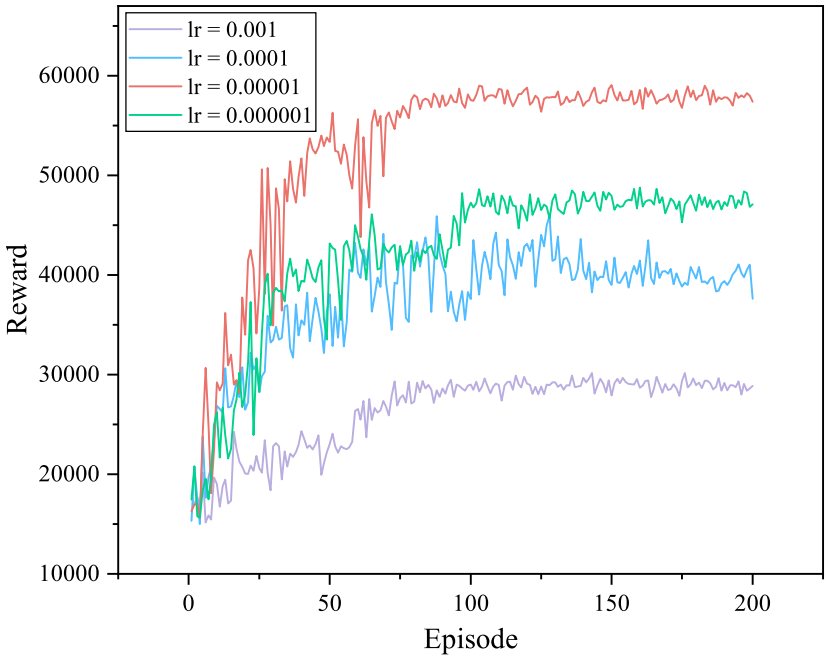

Fig. 3. Comparison of rewards with different learning rates in the DRL-DGSN algorithm.

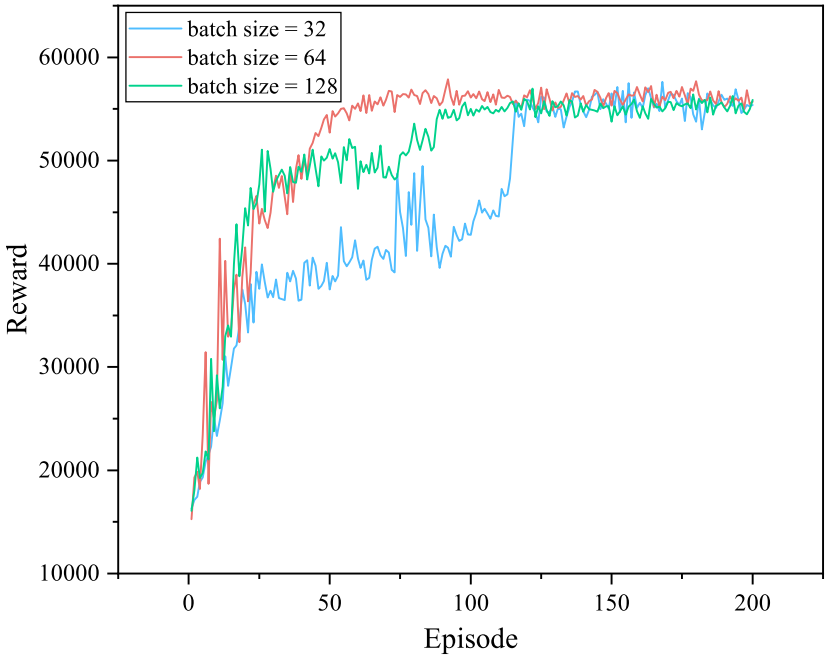

Fig. 4. Comparison of rewards for different batch sizes in the DRL-DGSN algorithm.

it circumvents the overestimation issue commonly associated with Q-values when a single network is utilized in standard DQN. These advantages empower the DRL-DGSN algorithm to more accurately perceive environmental changes and make optimal decisions, which enhance the overall convergence performance of the algorithm.

Fig. 7 illustrates the convergence of transmission rates for the worst-performing users under different algorithms. MDQN and UPRA prioritize maximizing total throughput, which causes them to favor users with better communication capabilities. This emphasis results in a decline in computation rates for the worst-performing users, which leads to difficulty in balancing fairness across the user base. On the contrary, DRL-DGSN incorporates user fairness constraints into the Bellman equation's objective function during the Q-network update process through a direct threshold-based penalty mechanism. When the transmission rate of the worst user falls below a predefined threshold, DRL-DGSN imposes a fixed negative reward signal to discourage actions that compromise fairness by utilizing a discretized reward and punishment function. Simultaneously, the dueling network architecture in DRL-DGSN decouples the state value function from the action advantage function. This procedure allows the Q-network to be more sensitive in identifying actions with "high global value but compromising fairness". The experimental results demonstrate that the total transmission rate of the worst-performing user in DRL-DGSN improves by 150.69% and 13.62% compared with those in MDQN and UPRA, respectively.

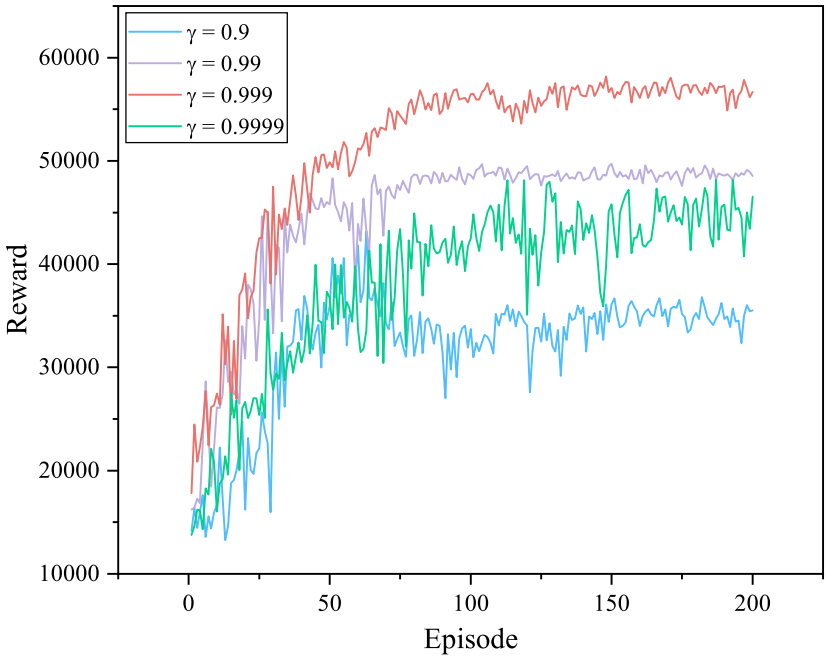

Fig. 5. Comparison of rewards with different discount factors in the DRL-DGSN algorithm.

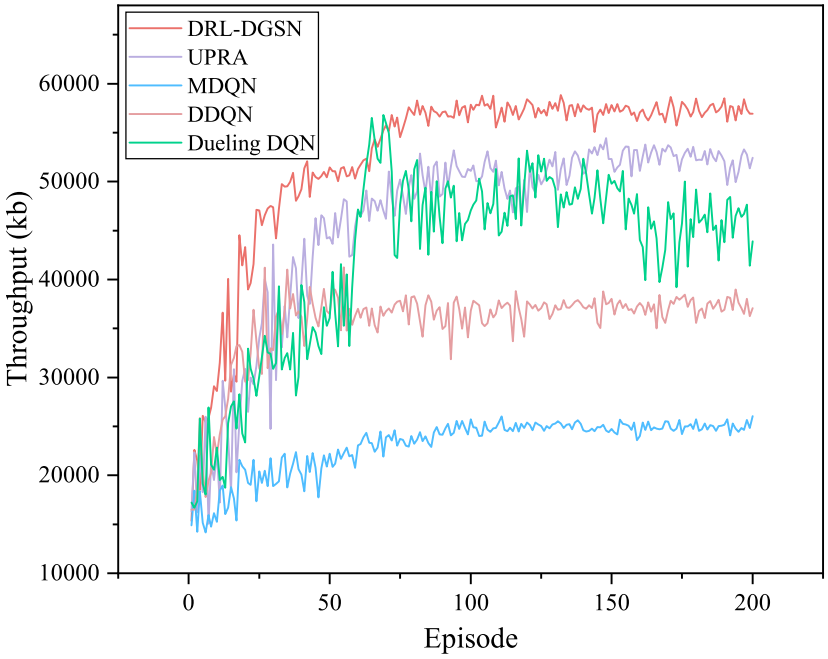

Fig. 6. Overall performance comparison.

6.4. Analysis of clustering strategies and update intervals Δ

As shown in Fig. 8, we examined the late-stage transmission rate performance of the DRL-DGSN under static clustering and different update intervals Δ (with the late stage defined as $t > 20$ s). The figure reveals that after a certain point, the total transmission rate gap between static clustering and periodic clustering (based on different update intervals Δ) gradually widens, with static clustering demonstrating a clear disadvantage. This occurs because static clustering fixes the association decisions between drones and users from the outset, failing to adapt to the environmental changes caused by user mobility. In contrast, periodic clustering allows the DRL-DGSN to dynamically optimize drone-user association strategies through regular reallocations, thereby continuously enhancing and sustaining high total transmission rate levels. Within the periodic clustering strategy, performance varies significantly across different update intervals. As Δ increases from 8 to 24, the median total transmission rate gradually decreases. This decline is due to excessively long update interval, causing cluster structure

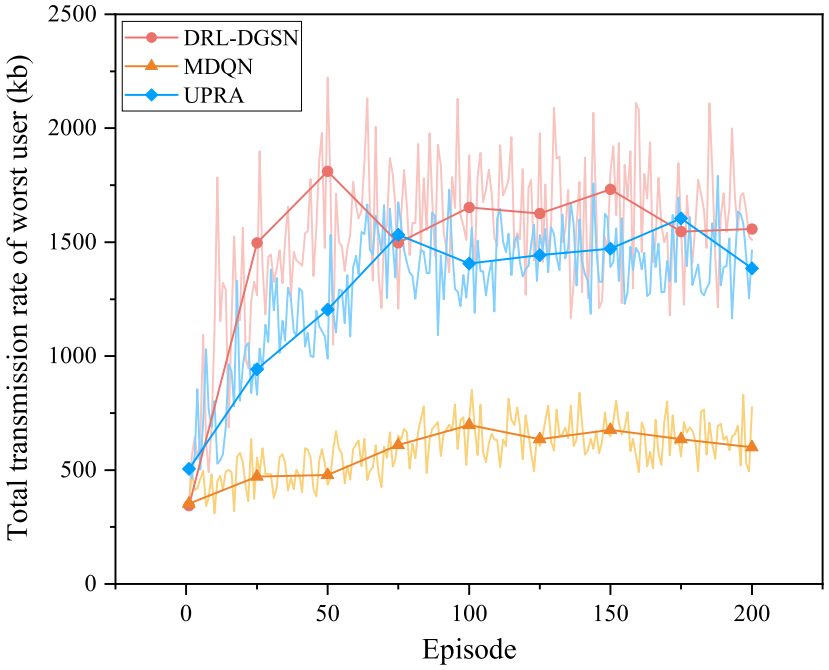

Fig. 7. Convergence of computation rate for worst users.

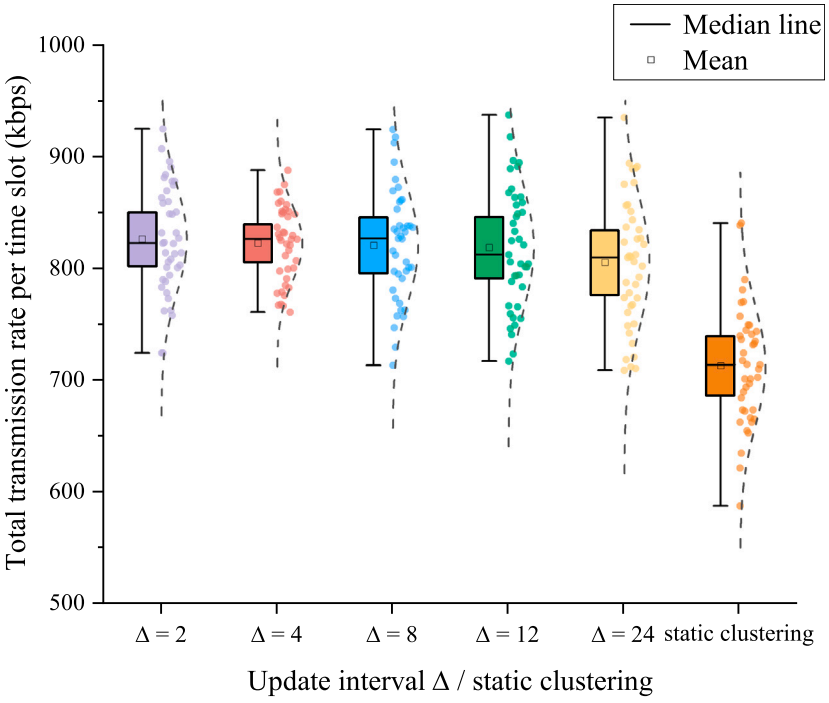

Fig. 8. Late-stage transmission rate of DRL-DGSN under static clustering and different update intervals Δ.

adjustments to lag behind dynamic user movements. For Δ ranging from 2 to 8, the overall total transmission rate levels are similar. However, the box plot for $\Delta = 4$ shows a 23.64% and 32.10% decrease compared to $\Delta = 2$ and $\Delta = 8$, respectively. This indicates that the DRL-DGSN exhibits lower total transmission rate fluctuations and significantly superior stability at the update interval $\Delta = 4$ compared to $\Delta = 2$ and $\Delta = 8$. In summary, the update interval $\Delta = 4$ achieves optimal stability while maintaining a high total transmission rate level. Therefore, we set the update interval for the DRL-DGSN to $\Delta = 4$.

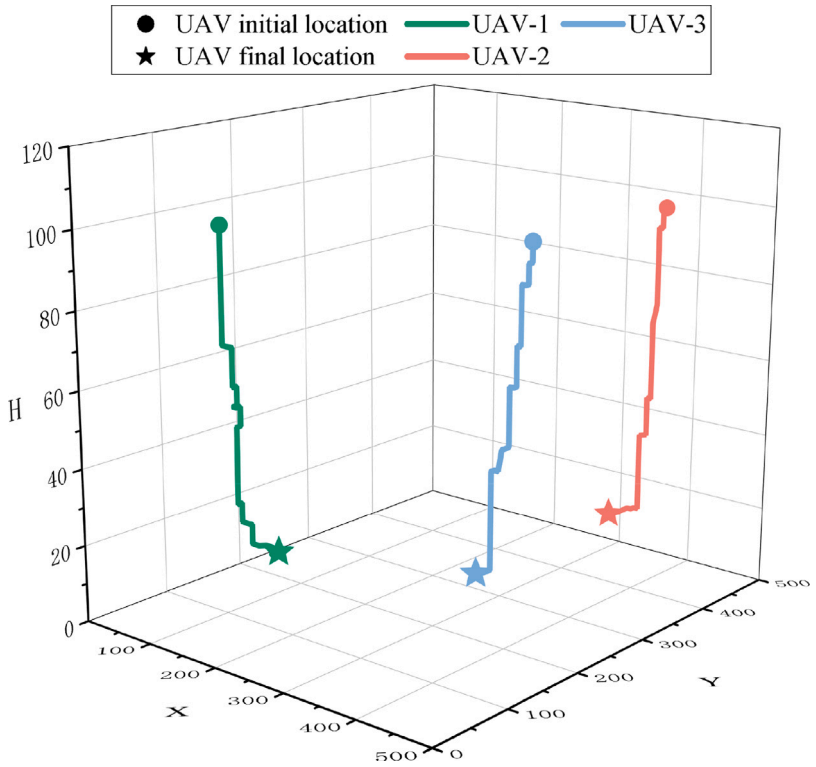

Fig. 9. Trajectory designed of UAVs.

6.5. Trajectory optimization for UAVs

Fig. 9 presents the trajectory designed of the UAV cluster optimized by the DRL-DGSN algorithm. Throughout the entire time slot T, the UAVs maintain a safe distance from one another, and their flight paths do not intersect. This observation validates the effectiveness of the user association strategy employed by the DUA-GMM algorithm. The DRL-DGSN algorithm leverages a mechanism of multi-agent collaboration, which enables real-time exchange of UAV position information via a shared network architecture. This feature facilitates collaborative obstacle avoidance in complex scenarios. The flight trajectories of all UAVs remain within the established motion boundaries, which demonstrates the effective boundary constraints imposed by DRL-DGSN. In addition, each UAV accurately identifies its associated user cluster and executes directional flight through dynamic trajectory adjustments. This capability arises from the trajectory control component of DRL-DGSN, which dynamically adjusts UAV trajectories by precisely estimating the Q-value of movements.

In summary, DRL-DGSN effectively achieves cooperative control over user cluster matching, UAV safety constraints, and dynamic trajectory adjustments through the integration of DUA-GMM algorithm, boundary constraint management, and multi-agent information sharing. Ultimately, this integration maximizes system throughput by simultaneously optimizing power allocation and trajectory planning for UAVs.

6.6. Effect of different numbers of users on the total system throughput

As the number of ground user increases, competition for resources among users intensifies, which significantly complicates the resource allocation algorithm. Accordingly, we consider a scenario with 3 UAVs and set the number of users for these UAVs to 3, 6, and 9, respectively. As illustrated in Fig. 10, the total throughput of the system initially exhibits a rapid increase before the growth rate begins to slow down as the number of users rises. Specifically, when the number of users increases from 3 to 6, a marked enhancement in total system throughput is observed. This result suggests that, within a certain range, increasing the number of users can lead to more efficient utilization of UAV resources, which improves overall throughput. However, when the number of users is further increased to 9, the growth rate of total system throughput diminishes. This slowdown can primarily be attributed to the limited computational power of the UAVs. As the number of users approaches a certain threshold, the UAV resources become nearly saturated, which results in a deceleration in the growth of total system throughput. Experimental results demonstrate that DRL-DGSN consistently maintains optimal performance across varying user sizes. In the scenario with 6 users, the total system throughput of DRL-DGSN improves by 9.61%, 149.52%, 25.59%, 53.31%, and 274.33% compared with those of UPRA, MDQN, DDQN, Dueling DQN, and Random, respectively.

6.7. Effect of different numbers of UAVs on the total system throughput

The number of UAVs directly affects the division of user clusters and the level of interference. In turn, this influence affects the parallel processing capabilities of the system. We have examined scenarios with 1, 2, and 3 UAVs to compare the system throughput of each algorithm. As illustrated in Fig. 11, the total system throughput increases across all cases as the number of UAVs rises. However, in the single-UAV scenario, the total system throughput decreases by only 22.15% compared with the scenario with 2 UAVs. This moderate increase can be attributed to that, in the single-UAV scenario, all users are grouped into a single service cluster managed by that UAV. This arrangement avoids the negative effects of inter-cluster interference, which results in a less significant reduction in total system throughput despite the fewer UAVs available. Experimental results indicate that DRL-DGSN achieves the highest total system throughput across different numbers of UAVs. This superior performance is attributable to the integration of the DUA-GMM algorithm within DRL-DGSN, which effectively models user distribution and constrains cluster capacity through probabilistic methods. This capability significantly reduces inter-cluster interference in multi-UAV scenarios. In addition, the positions and channel state

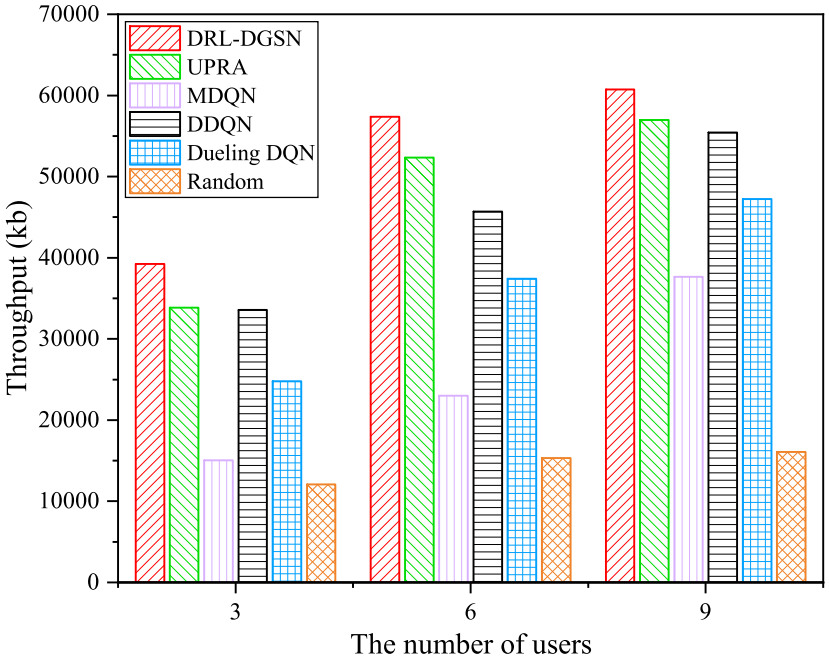

Fig. 10. Effect of different numbers of users on the total system throughput.

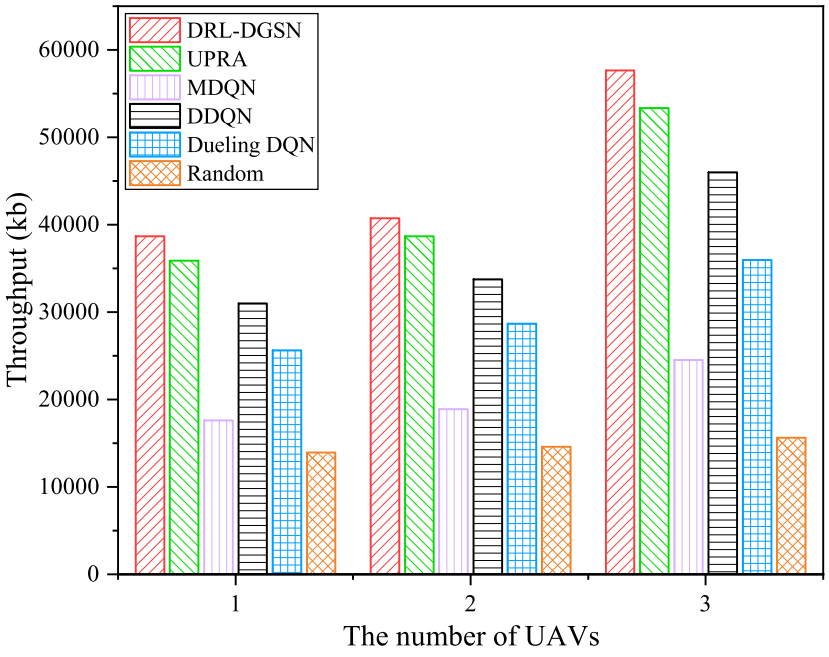

Fig. 11. Effect of different numbers of UAVs on the total system throughput.

information of UAVs are modeled as the state space of MDP, and the network parameters are shared among the agents. This consideration greatly improves the consistency of resource allocation among UAV clusters. In the multi-UAV cooperative scenario with 3 UAVs, DRL-DGSN enhances the total system throughput by 8.04%, 135.00%, 25.38%, 60.28%, and 269.32% compared with UPRA, MDQN, DDQN, Dueling DQN, and Random, respectively.

6.8. Effect of different maximum transmit power values of the UAV on total system throughput

This experiment investigates scenarios where the maximum transmit power values of the UAV are set to 20, 25, and 30 dBm, respectively. As illustrated in Fig. 12, the gradual increase in the maximum transmit power of the UAV also raises the power allocated to each user. Given the positive correlation between the power allocated to a user and the resulting throughput, the total system throughput demonstrates a consistent increase across different algorithms. The experimental results indicate that DRL-DGSN achieves optimal performance metrics across various maximum UAV transmit power scenarios. Specifically, when the maximum transmit power of the UAV is 20 dBm, the total system throughput of DRL-DGSN improves by 22.48%, 148.96%, 26.88%, 69.39%, and 248.11% compared with those of UPRA, MDQN, DDQN, Dueling DQN, and Random, respectively. This advantage arises from the synergistic optimization of NOMA technology and DRL within DRL-DGSN. By dynamically optimizing the power allocation strategy for

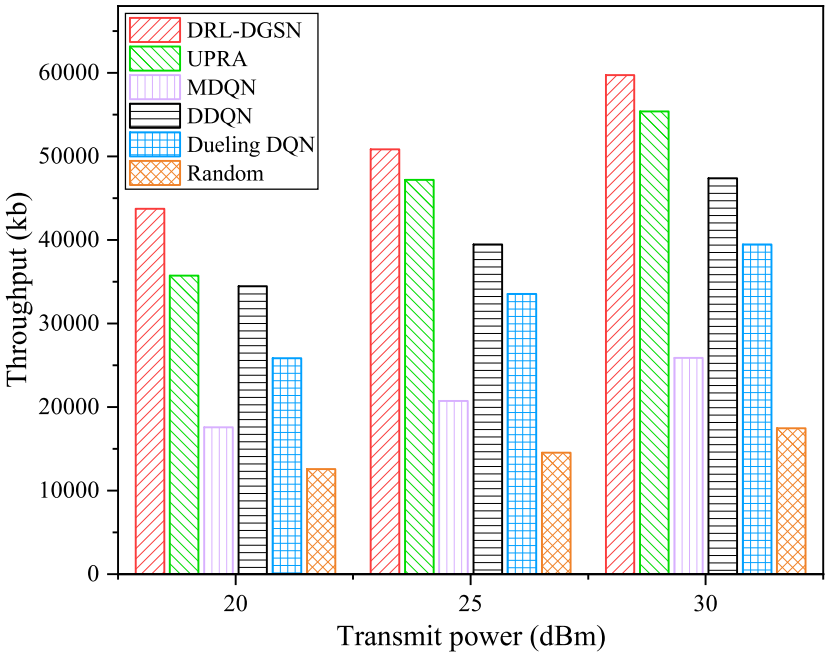

Fig. 12. Effect of different maximum transmit power values of the UAV on the total system throughput.

the UAV through the Dueling network architecture and employing SIC technology to mitigate intra-cluster interference, DRL-DGSN maximizes cumulative system throughput in power-constrained scenarios.

7. Conclusion

In this study, a NOMA-assisted AEC network architecture is developed, and user association, UAV trajectory, and power allocation strategies are simultaneously optimized with the goal of maximizing cumulative system throughput. To address the non-convexity of the optimization problem, it is modeled as an MDP. Meanwhile, the DRL-DGSN algorithm is proposed based on the DRL framework, offering an effective solution to the resource allocation challenges in AEC. First, users are dynamically clustered according to their spatial distribution using the DUA-GMM algorithm. As a result, the optimal association strategy between user clusters and UAVs is established. Subsequently, a DDQN algorithm framework based on a dueling network architecture is introduced to achieve trajectory control and power allocation of UAVs. This approach effectively addresses the issue of Q-value overestimation and enhances the generalization ability of decision making by decoupling the state value function from the action advantage function. The experimental results demonstrate that improvements in system throughput are achieved by the DRL-DGSN algorithm, with increases of 129.66%, 10.06%, 25.85%, 49.10%, and 238.04% compared with those of MDQN, UPRA, DDQN, Dueling DQN, and Random, respectively. It also exhibits superior converge to the compared algorithms. Additionally, DRL-DGSN demonstrates considerable practical potential for emergency communication scenarios, where autonomous UAV collaboration can substantially enhance network service performance.

However, DRL-DGSN still exhibits certain limitations. Its performance relies on a set of idealized assumptions, such as the full availability of channel state information, user mobility strictly adhering to a random walk model, and the absence of hardware constraints for UAVs. These assumptions deviate considerably from the conditions encountered in real-world scenarios. Experimental validation scenarios are restricted to small-scale networks and do not address large-scale deployment requirements. Furthermore, the optimization objective is limited to system throughput, which makes it challenging to accommodate diverse service requirements.

To address these challenges, future work will focus on developing a multi-objective optimized DRL model to address additional performance metrics such as delay, energy consumption, and task completion rate, in order to accommodate heterogeneous service requirements. Furthermore, a MADRL resource allocation scheme within a federated learning framework will be explored to enhance the algorithm's adaptability and scalability in large-scale network environments.

CRediT authorship contribution statement

Longxin Zhang: Writing – review & editing, Writing – original draft, Methodology, Funding acquisition, Formal analysis. **Xiaotong Lu:** Writing – original draft, Data curation. **Jing Liu:** Writing – original draft, Software. **Yanfen Zhang:** Software, Data curation. **Jianguo Chen:** Validation, Methodology. **Buqing Cao:** Methodology. **Keqin Li:** Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank thirteen anonymous reviewers for their suggestions to improve the manuscript. This work was partially funded by the Natural Science Foundation of Hunan Province, China (Grant Nos. 2023JJ50204), the Scientific Research Foundation of Hunan Provincial Education Department, China (Grant No. 23B0560), the National Key R&D Program of China (Grant No. 2018YFB1003401), the National Natural Science Foundation of China (Grant Nos. 62572186, 61702178, 62072172, 62002110, 62372486), and the Open Project Funding of the Key Laboratory of Intelligent Sensing System and Security (Hubei University), Ministry of Education.

Data availability

The authors do not have permission to share data.

References

[1] C. Park, J. Lee, Mobile edge computing-enabled heterogeneous networks, IEEE Trans. Wirel. Commun. 20 (2) (2021) 1038–1051.

[2] U. Awada, J. Zhang, S. Chen, S. Li, S. Yang, Resource-aware multi-task offloading and dependency-aware scheduling for integrated edge-enabled IoV, J. Syst. Archit. 141 (2023) 102923.

[3] L. Zhang, R. Tan, M. Ai, H. Xiang, C. Peng, Z. Zeng, DSUTO: Differential rate SAC-based UAV-assisted task offloading algorithm in collaborative edge computing, in: 2023 IEEE 29th International Conference on Parallel and Distributed Systems, ICPADS, 2023, pp. 2328–2335.

[4] X. Yang, L. Shao, Green and efficient path planning framework based on multi-granularity view fusion for amphibious unmanned aerial vehicles, Eng. Appl. Artif. Intell. 151 (2025) 110643.

[5] A. Mohajer, M. Sam Daliri, A. Mirzaei, A. Ziaeddini, M. Nabipour, M. Bavaghar, Heterogeneous computational resource allocation for NOMA: Toward green mobile edge-computing systems, IEEE Trans. Serv. Comput. 16 (2) (2023) 1225–1238.

[6] Y. Prathyusha, T.-L. Sheu, Resource allocations for coexisting eMBB and URLLC services in multi-UAV aided communication networks for cellular offloading, IEEE Trans. Veh. Technol. 73 (5) (2024) 6658–6671.

[7] V. Melnykov, S. Michael, Clustering large datasets by merging K-means solutions, J. Classification 37 (1) (2020) 97–123.

[8] E. Hancer, B. Xue, M. Zhang, A survey on feature selection approaches for clustering, Artif. Intell. Rev. 53 (6) (2020) 4519–4545.

[9] S. Shen, G. Shen, X. Yang, F. Xia, H. Du, X. Kong, Mean-field reinforcement learning for decentralized task offloading in vehicular edge computing, J. Syst. Archit. 146 (2024) 103048.

[10] S.A. Huda, S. Moh, Survey on computation offloading in UAV-Enabled mobile edge computing, J. Netw. Comput. Appl. 201 (2022) 103341.

[11] L. Zhang, R. Tan, Y. Zhang, J. Peng, J. Liu, K. Li, UAV-assisted dependency-aware computation offloading in device-edge-cloud collaborative computing based on improved actor-critic DRL, J. Syst. Archit. 154 (2024) 103215.

[12] X. Chen, J. Zhang, B. Lin, Z. Chen, K. Wolter, G. Min, Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments, IEEE Trans. Parallel Distrib. Syst. 33 (3) (2022) 683–697.

[13] X. Chen, S. Hu, C. Yu, Z. Chen, G. Min, Real-time offloading for dependent and parallel tasks in cloud-edge environments using deep reinforcement learning, IEEE Trans. Parallel Distrib. Syst. 35 (3) (2024) 391–404.

[14] L. Liao, Y. Lai, F. Yang, W. Zeng, Online computation offloading with double reinforcement learning algorithm in mobile edge computing, J. Parallel Distrib. Comput. 171 (2023) 28–39.

[15] A. Sadiki, J. Bentahar, R. Dssouli, A. En-Nouaary, H. Otrok, Deep reinforcement learning for the computation offloading in MIMO-based edge computing, Ad Hoc Netw. 141 (2023) 103080.

[16] S. Pang, T. Wang, H. Gui, X. He, L. Hou, An intelligent task offloading method based on multi-agent deep reinforcement learning in ultra-dense heterogeneous network with mobile edge computing, Comput. Netw. 250 (2024) 110555.

[17] M. Saberikia, H. Farbeh, M. Fazeli, Improving energy efficiency and fault tolerance of mission-critical cloud task scheduling: A mixed-integer linear programming approach, Sustain. Comput.: Inform. Syst. 45 (2025) 101068.

[18] G. Wu, X. Chen, Z. Gao, H. Zhang, S. Yu, S. Shen, Privacy-preserving offloading scheme in multi-access mobile edge computing based on MADRL, J. Parallel Distrib. Comput. 183 (2024) 104775.

[19] X. Li, X. Du, N. Zhao, X. Wang, Computing over the sky: Joint UAV trajectory and task offloading scheme based on optimization-embedding multi-agent deep reinforcement learning, IEEE Trans. Commun. 72 (3) (2024) 1355–1369.

[20] H. Hao, C. Xu, W. Zhang, S. Yang, G.-M. Muntean, Joint task offloading, resource allocation, and trajectory design for multi-UAV cooperative edge computing with task priority, IEEE Trans. Mob. Comput. 23 (9) (2024) 8649–8663.

[21] Q. Luo, T.H. Luan, W. Shi, P. Fan, Deep reinforcement learning based computation offloading and trajectory planning for multi-UAV cooperative target search, IEEE J. Sel. Areas Commun. 41 (2) (2023) 504–520.

[22] H. Li, H. Li, C. Zhang, D. Li, X. Duan, J. Liu, A UAV-assisted dynamic offloading based on maximum clique algorithm with weighted graphs in mobile edge computing, Comput. Netw. 258 (2025) 111028.

[23] S. Ghosh, P. Kuila, M. Bey, M. Azharuddin, Quantum-inspired gravitational search algorithm-based low-price binary task offloading for multi-users in unmanned aerial vehicle-assisted edge computing systems, Expert Syst. Appl. 263 (2025) 125762.

[24] J. Yan, X. Zhao, Z. Li, Deep-reinforcement-learning-based computation offloading in UAV-assisted vehicular edge computing networks, IEEE Internet Things J. 11 (11) (2024) 19882–19897.

[25] M. Hadi, R. Ghazizadeh, Joint resource allocation, user clustering and 3-D location optimization in multi-UAV-enabled mobile edge computing, Comput. Netw. 218 (2022) 109420.

[26] A. Umar, S.A. Hassan, H. Jung, S. Garg, G. Kaddoum, M.S. Hossain, HAC-SAGIN: High-altitude computing enabled space-air-ground integrated networks for 6G, Comput. Netw. 254 (2024) 110797.

[27] W. Lu, Y. Ding, Y. Gao, Y. Chen, N. Zhao, Z. Ding, A. Nallanathan, Secure NOMA-based UAV-MEC network towards a flying eavesdropper, IEEE Trans. Commun. 70 (5) (2022) 3364–3376.

[28] Y. Xu, T. Zhang, D. Yang, Y. Liu, M. Tao, Joint resource and trajectory optimization for security in UAV-assisted MEC systems, IEEE Trans. Commun. 69 (1) (2021) 573–588.

[29] X. Dai, Z. Lu, X. Chen, X. Xu, F. Tang, Multiagent RL-based joint trajectory scheduling and resource allocation in NOMA-assisted UAV swarm network, IEEE Internet Things J. 11 (8) (2024) 14153–14167.

[30] R. Zhong, X. Liu, Y. Liu, Y. Chen, Multi-agent reinforcement learning in NOMA-aided UAV networks for cellular offloading, IEEE Trans. Wirel. Commun. 21 (3) (2022) 1498–1512.

[31] X. Luo, J. Xie, L. Xiong, Z. Wang, Y. Liu, UAV-assisted fair communications for multi-pair users: A multi-agent deep reinforcement learning method, Comput. Netw. 242 (2024) 110277.

[32] X. Xu, K. Liu, P. Dai, F. Jin, H. Ren, C. Zhan, S. Guo, Joint task offloading and resource optimization in NOMA-based vehicular edge computing: A game-theoretic DRL approach, J. Syst. Archit. 134 (2023) 102780.

[33] T.-H. Nguyen, T.P. Truong, A.-T. Tran, N.-N. Dao, L. Park, S. Cho, Intelligent heterogeneous aerial edge computing for advanced 5G access, IEEE Trans. Netw. Sci. Eng. 11 (4) (2024) 3398–3411.

[34] M.H. Naim Shaikh, A. Celik, A.M. Eltawil, G. Nauryzbayev, Grant-free NOMA through optimal partitioning and cluster assignment in STAR-RIS networks, IEEE Trans. Wirel. Commun. 23 (8) (2024) 10166–10181.

[35] X. Zhou, J. Xiong, H. Zhao, X. Liu, B. Ren, X. Zhang, J. Wei, H. Yin, Joint UAV trajectory and communication design with heterogeneous multi-agent reinforcement learning, Sci. China Inf. Sci. 67 (3) (2024) 132302.

[36] L. Liu, T. Jing, W. Li, J. Duan, W. Mao, H. Liu, G. Liu, Meta learning-based deep reinforcement learning algorithm for task offloading in dynamic vehicular network, Eng. Appl. Artif. Intell. 143 (2025) 109966.

[37] N. Lin, H. Tang, L. Zhao, S. Wan, A. Hawbani, M. Guizani, A PDDQNLP algorithm for energy efficient computation offloading in UAV-assisted MEC, IEEE Trans. Wirel. Commun. 22 (12) (2023) 8876–8890.

[38] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, D. Niyato, Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing, IEEE Trans. Wirel. Commun. 21 (9) (2022) 6949–6960.

[39] X. Yang, D. Qin, J. Liu, Y. Li, Y. Zhu, L. Ma, Deep reinforcement learning in NOMA-assisted UAV networks for path selection and resource offloading, Ad Hoc Netw. 151 (2023) 103285.

[40] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, Proc. AAAI Conf. Artif. Intell. 30 (1) (2016).

[41] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, N. De Freitas, Dueling network architectures for deep reinforcement learning, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML '16, JMLR.org, 2016, pp. 1995–2003.

[42] L. Zhang, R. Tan, B. Cao, L.A.K. Li, K. Li, EP-MUSTO: Entropy-enhanced DRL-based task offloading in secure multi-UAV-assisted collaborative edge computing, IEEE Internet Things J. 12 (16) (2025) 34459–34470.