UAV-Enabled Semantic Communication in Mobile Edge Computing Under Jamming Attacks: An Intelligent Resource Management Approach

Shuai Liu, *Graduate Student Member, IEEE*, Helin Yang, *Senior Member, IEEE*, Mengting Zheng, *Student Member, IEEE*, Liang Xiao, *Senior Member, IEEE*, Zehui Xiong, *Senior Member, IEEE*, and Dusit Niyato, *Fellow, IEEE*

***Abstract*—The integration of semantic communication with mobile edge computing (MEC) has emerged as a prominent research area. In this paper, we explore a novel scenario where semantic communication is integrated with unmanned aerial vehicles (UAVs) to enhance MEC, particularly in the face of jamming attacks. Our research focuses on addressing the resource management challenge to minimize task completion time and maximize semantic spectral efficiency (SSE) while adhering to quality of service requirements and resource constraints. Given the non-convexity of this problem and the dynamic behavior of jamming attacks, this paper proposes a deep reinforcement learning (DRL) algorithm by jointly optimizing UAV trajectories, user associations, and channel selections against jamming. In detail, the proposed anti-jamming DRL-based resource management approach can effectively capture the jammer's behavior, and learn to adjust semantic task and resource scheduling strategies with the objective to minimize the negative effect of jamming attacks on task offloading and semantic communication. Simulation results demonstrate that the proposed approach outperforms baseline algorithms in terms of task completion time and total SSE under different real-world settings.**

***Index Terms*—Semantic communication, unmanned aerial vehicle, mobile edge computing, deep reinforcement learning, resource management, anti-jamming.**

Manuscript received 22 January 2024; revised 30 May 2024; accepted 23 August 2024. Date of publication 11 September 2024; date of current version 13 November 2024. This work was supported in part by the Fujian Provincial Natural Science Foundation of China under Grant 2024J09002; in part by the National Natural Science Foundation of China under Grant 62371408, Grant 62301467, and Grant U21A20444; in part by the Fundamental Research Funds for the Central Universities under Grant 20720220080; in part by the Natural Science Foundation of Xiamen China under Grant 3502Z202371010; in part by the Xiaomi Young Talents Program, the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research and Development Program, Defence Science Organisation (DSO) National Laboratories under the AI Singapore Program under Grant FCP-NTU-RG-2022-010 and Grant FCP-ASTAR-TG-2022-003; in part by Singapore Ministry of Education (MOE) Tier 1 under Grant RG87/22; in part by Nanyang Technological University (NTU) Centre for Computational Technologies in Finance (NTU-CCTF); in part by Singapore University of Technology and Design (SUTD) under Grant SRG-ISTD-2021-165; in part by SUTD-Zhejiang University (ZJU) IDEA under Grant SUTD-ZJU (VP) 202102; in part by the Ministry of Education, Singapore, under its Singapore Management University (SMU)-SUTD Joint Grant under Award 22-SIS-SMU-048; and in part by the SUTD Kickstarter Initiative under Grant SKI 20210204. An earlier version of this paper was presented in part at the IEEE/CIC International Conference on Communications in China, Hangzhou, China, in August 2024. The associate editor coordinating the review of this article and approving it for publication was F. Hou. *(Corresponding author: Helin Yang.)*

Shuai Liu and Mengting Zheng are with the School of Informatics and the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Xiamen University, Xiamen 361005, China (e-mail: liushuai@stu.xmu.edu.cn).

Helin Yang and Liang Xiao are with the School of Informatics, the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, and the Institute of Artificial Intelligence, Xiamen University, Xiamen 361005, China (e-mail: helinyang066@xmu.edu.cn; lxiao@xmu.edu.cn).

Zehui Xiong is with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372 (e-mail: zehui_xiong@sutd.edu.sg).

Dusit Niyato is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: dniyato@ntu.edu.sg).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2024.3454073.

Digital Object Identifier 10.1109/TWC.2024.3454073

I. Introduction

IN THE era of rapid development in 5th generation mobile networks (5G) and 6th generation mobile networks (6G) communications, the combination of artificial intelligence (AI) technology with wireless communication became a growing trend [1], [2]. Even though this trend brought a lot of convenience to people's lives [3], [4], due to the rapid growth in the number of wireless communication users over a short period, communication and computation resources could not keep up with the demands of wireless communication. This resulted in resource scarcity in wireless communications, making it challenging to achieve optimal communication quality. Semantic communication technology could address the issue of insufficient spectrum resources [5], [6]. Semantic communication technology referred to a technology of wireless communications that extracted the meaning of data [7]. This technology could extract relevant information from tasks while eliminating unnecessary data, thereby improving communication efficiency and increasing the volume of communication tasks. This generated significant interest among researchers in recent years [5], [6], [7].

DeepSC [8], an advanced semantic communication system based on deep learning, was introduced. Diverging from conventional communication systems, this system maximized semantic capacity by recovering sentence meanings while minimizing semantic errors, allowing its adaptation to various environments through machine learning techniques. This work was one of the pioneering works in the field of semantic communication. In [9], the authors presented an AI-based semantic communication approach; this approach addressed the previously undiscussed semantic importance of features for

AI tasks in earlier research, establishing a prototype for surface defect detection using semantic communication methods with AI tasks (AITs), which resulted in a 40% improvement in classification accuracy. Another notable development, discussed in [10], involved the design of a speech communication system for voice signals. This system outperformed traditional communication in various speech signal metrics, including distortion ratio and perceived speech distortion evaluation. This model made significant contributions to the research in the field of speech semantic communication, improving the efficiency of speech semantic communication. In [11] and [12], a novel semantic perception network resource allocation model combining semantic communication and a resource allocation scheme combining semantic communication with the internet of vehicles were proposed, respectively, and they enhanced user satisfaction and other aspects. Traditional communication transmission schemes sent all data to the receiver without processing, inevitably transmitting a large amount of irrelevant data during the transmission process, leading to insufficient spectrum resources and reduced communication efficiency [13]. The aforementioned semantic communication schemes, compared to traditional communication schemes, transmitted only relevant and useful information [14], [15]. By ensuring that the receiver obtained relevant useful information while saving spectrum resources and enhancing spectrum utilization, the system's communication performance was improved.

However, the limited availability of computational resources emerged as a significant factor contributing to the degradation of communication quality. Recognizing that semantic communication alone cannot adequately address this issue, edge computing technology [16] was introduced. Related work is as follows.

A. Related Work

In recent years, there had been investigations into the integration of semantic communication with edge computing. The authors in [17] had researched how semantic communication enhanced edge intelligence, and then analyzed edge-driven voice communication, showcasing the advantages of voice communication and edge intelligence technology to people. In the study [18], an incentive mechanism based on deep learning auctions for the allocation of computational resources in a meta-space supported by semantic communication was designed, increasing the efficiency of edge computing resource allocation. In the work [19], a semantic caching model for edge computing systems was proposed. This model cached domain-specific common models and user-specific models, reducing the time and resources required to establish individual KBs, while accurately capturing the semantics behind user messages to enhance the effectiveness of semantic communication. It had become evident from [17], [18], and [19] that combining semantic communication with edge computing was highly feasible. Semantic communication could optimize communication resources but might not have optimized computing resources. Integrating edge computing had allowed for the simultaneous optimization of both computational and communication resources [20]. This was because edge computing could process and store data at the network edge, reducing the system's computational load. As edge devices were closer to users, they could further reduce the bandwidth needed to transmit the same data, thereby alleviating the pressure on communication resources [21] and enhancing the efficiency of semantic communication. Therefore, incorporating edge computing technology in semantic communication had been very promising. However, due to limitations imposed by the environment, users, and edge devices themselves, ground-based edge devices could not provide users with optimal communication services. Therefore, unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) technology had emerged [22].

UAVs possessed excellent flexibility and were less affected by ground environments [23], [24], making them ideal for assisting in MEC and mitigating the reduction in communication quality caused by obstacles obstructing or preventing the installation of ground-edge devices. In [25] and [26], the traditional optimization algorithm was used to optimize the UAV-assisted MEC strategy, which reduced the delay and solved the security problem in the unloading process. While [25], [26] achieved favorable results, they mainly employed traditional algorithms for UAV-assisted MEC. Given the increasing complexity of communication environments at the time, reinforcement learning (RL) algorithms could better adapt to dynamic and complex communication scenarios compared to traditional methods. RL allowed dynamic interaction with the environment, autonomous learning in dynamic settings, and the accumulation of experience for greater rewards [27]. The literature [28], [29], [30] also used the deep reinforcement learning (DRL) algorithm to optimize the UAV-assisted MEC strategy and achieved good results in reducing power consumption and task backlog. While these UAV-assisted MEC solutions demonstrated impressive performance at the time, few studies had combined UAV-assisted MEC with semantic communication. Semantic communication could improve information transfer speed and reduce latency. In previous UAV-assisted MEC schemes [25], [26], [27], [28], [29], [30], traditional wireless communication methods had been mainly used. Users, UAVs, and base stations (BSs) transmitted raw data directly to each other without preprocessing. This approach had led to the transmission of excessive data volumes, potentially occupying too much spectrum resources and causing reduced communication efficiency. Additionally, UAVs had limited battery capacity, and transmitting or receiving large volumes of data had resulted in excessive energy consumption, preventing UAVs from fully leveraging their advantages [31]. By integrating semantic communication technologies with UAV-assisted MEC, the flexibility of UAVs, line-of-sight (LoS) channels, and computing resources had been effectively utilized [32], reducing communication energy consumption, enhancing data transmission efficiency, and significantly improving communication system performance. Additionally, communication environments had been vulnerable to jamming attacks from malicious jammers, making it crucial to have studied how to safeguard the benefits of UAV-assisted MEC in a jamming environment.

Many studies had investigated communication resilience to jamming [33], [34], [35], laying the groundwork for UAV-assisted MEC in a jamming environment [36], [37], [38]. In [33], a UAV-based anti-jamming video transmission solution using RL was introduced. This approach optimized video quantization parameters, channel coding rates, modulation, and attack control strategy selections. It enabled UAVs to guarantee video quality without relying on video interference or business models while reducing energy consumption. This work significantly improved video quality, reduced energy consumption, and latency. Regarding UAV-assisted MEC in a jamming environment, [36] presented a secure MEC system with UAV optimization. This system jointly optimized UAV positioning, user transmission power, UAV jamming power, offloading ratio, and UAV computational capabilities. It demonstrated a fundamental trade-off between security and latency, ultimately improving the efficiency of UAV-assisted MEC. While the UAV-assisted MEC solutions in the jamming environment [33], [34], [35], [36], [37], [38] had shown promising results, they had not integrated semantic communication technology. Combining semantic communication technology with UAV-assisted MEC could fully leverage the flexibility and easy deployment advantages of UAVs, improve communication efficiency and channel stability, and effectively mitigate the impact of jamming attacks. Therefore, there had still been room for improving system benefits.

B. Motivation and Main Contribution

In our work, we discuss a UAV-assisted MEC network that incorporates semantic communication in a jamming environment. In this network, ground users generate tasks, and these tasks use semantic communication to extract and transmit the most meaningful and useful information to UAVs performing MEC tasks. The optimization objective is to minimize the system's task completion time by jointly optimizing actions such as UAV trajectories, user associations, and offloading channel selections, given constraints on computational and channel resources. Specifically, we employ distinct DRL algorithms for decision-making by UAVs and users, aiming to maximize the system's reward in dynamic environments. The primary contributions of the paper are summarized as follows.

- To solve the resource allocation problem related to minimizing task completion time and maximizing total SSE, our work proposes an approach that combines semantic communication and UAV-assisted MEC under jamming attacks. To the best of our knowledge, this is the first work on integrating semantic communication with UAV-assisted MEC against jamming. As the problem is non-convex with NP-hard complexity involving time-series mixed integers, we then design and implement a combination algorithm, Twin Delayed Deep Deterministic Policy Gradient and Double Deep Q-Network (T5D), to jointly optimize the semantic tasks, UAV trajectories, user associations, and offloading channel selections.
- In the model, UAVs' actions are continuous, while users' and jammers' actions are discrete. Consequently, this work treats UAVs, users, and jammers as separate intelligent agents, each making its own decisions. This paper applies the TD3 algorithm for the strategic decision-making of UAVs, and the DDQN algorithm for the strategic decision-making of users, to maximize the system's benefits. Furthermore, the paper considers jammers as intelligent agents, employing the DQN algorithm for jammer's strategy decisions. This strengthens jamming strategies, enabling the algorithm to learn in adverse jamming environments and enhancing its performance in practical scenarios.
- The algorithm jointly optimizes the strategies of UAV and users in jamming environments. Through simulations and comparisons with different algorithms, the paper demonstrates that the proposed algorithm achieves maximum benefits under various scenarios, including different average number of semantic symbols K, and different maximum jamming power. Compared to the benchmark algorithms, the total SSE of the proposed algorithm is improved at least by approximately 4.00%. In comparison to non-semantic communication algorithms, the proposed algorithm achieves a reduction in task completion time of about 27.18%. These data comprehensively demonstrate the effectiveness of the proposed algorithm in enhancing UAV-assisted MEC strategies and semantic communication performances against jamming.

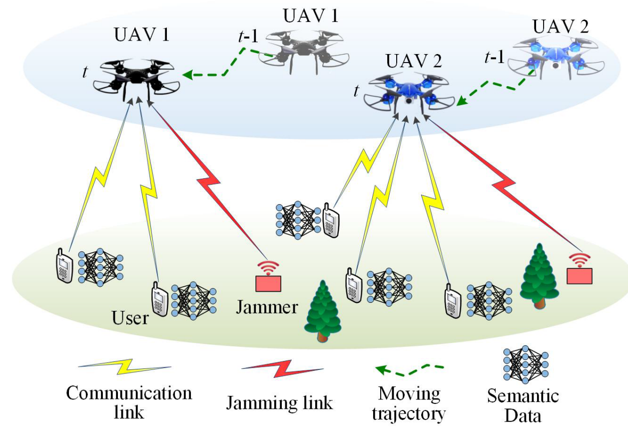


Fig. 1. A UAV-assisted MEC for semantic communication system under jamming attacks.

The rest of this paper is organized as follows. Section II presents the system model and problem formulation. Section III introduces the RL algorithms relevant to the optimization problem. Section IV details the specific solution algorithms for the optimization problem. Section V showcases the performance of the proposed algorithm through simulation results. Finally, Section VI presents the conclusions.

II. System Model and Problem Formulation

This paper investigates a UAV-assisted MEC system incorporating semantic communication under jamming attacks, as illustrated in Fig. 1. In certain communication scenarios, limitations in channel resources, users' computational capabilities, the coverage range of the base station (BS), coupled with interference from jammers, lead to issues such as excessive task completion time. Introducing UAV-assisted MEC can effectively address the issues. In this system, UAVs are introduced as aerial BSs, stationed at fixed altitudes to assist ground BSs in receiving and processing tasks from ground users, thereby assisting MEC. Then, the paper employs pre-trained semantic encoders and decoders to facilitate semantic communication. The DeepSC model from [8] is utilized to reduce the cost of offloading tasks from users to UAVs and minimize task completion time for users in jamming environments. In this paper, during the process of offloading tasks from users to UAVs, it is assumed that the tasks being transmitted are similar to those in [8]. The encoding results of the DeepSC model are directly used, combined with relevant information such as SINR, to derive variables related to semantic communication, thus eliminating the need for additional data compression and storage by the DeepSC model. Subsequently, the study focuses on the data transmission process post-encoding, with the aim of optimizing the communication transmission latency and computation latency of the tasks, hence the data processing and compression demands of the DeepSC model are not discussed. Our system model is general and other semantic encoder and decoder can be adopted.

As shown in Fig. 1, the system comprises N UAVs, N intelligent dynamic jammers, and M ground users. The sets of UAVs, jammers, and users are denoted as $U^{uav} = \{u_1^{uav}, u_2^{uav}, \cdots, u_N^{uav}\}$, $J = \{j_1, j_2, \cdots, j_N\}$, and $U^{us} = \{u_1^{us}, u_2^{us}, \cdots, u_M^{us}\}$, respectively. At time slot t, the task load of user m is denoted as $s_{t,m}$. In this system, the entire process of UAV task execution is divided into equal intervals of T time slots, $t = \{1, 2, \cdots, T\}$, each with a fixed duration of T^s. At time slot t, each user selects a UAV and offloads tasks to the hovering UAV. The system has C channels, denoted as $c = \{1, 2, \cdots, C\}$. In each time slot t, an user selects one channel from the available channels to unload tasks to the chosen UAV. The task offloading process for users consists of three steps. Firstly, at time slot t, the task of the m-th user U_m^{us} is encoded using the DeepSC model [20], transforming it into a semantic task $s_{t,m} = \left\{s_{t,m}^d, s_t^l, s_t^v\right\}$ transmitted to the corresponding UAV. Here, $s_{t,m}^d$ refers to the number of sentences offloaded to the UAV at time slot t, s_t^l is the average hardware requirement for each sentence at time slot t, and s_t^v represents the size of each sentence at time slot t. Similar to the literature [20], the task size depends primarily on $s_{t,m}^d$. In this work, it is assumed that $s_{t,m}^l$ and $s_{t,m}^v$ for each user are constant, does not change over time. Subsequently, these semantic tasks are decoded using the DeepSC model on the UAV for computation. Finally, the computed results are returned to the users. Similar to [39], the calculation results are overlooked due to their relatively small scale. The users move randomly within a small range or remain stationary in each time slot, while UAVs fly to assist task execution until the end of T time slots. Moreover, each intelligent jammer transmits jamming power to interfere with task unloading towards a UAV.

A. Task Processing Model

The task processing model is illustrated in Fig. 2. Each UAV at time slot t is divided into two phases: the flying phase and the hovering phase. During the flying phase, the UAV moves from one location to another, while during the hovering phase, the UAV hovers at a designated point to receive task offloading from users and compute the tasks. At the initial time slot t, users commence by selecting a UAV for semantic task offloading, followed by the choice of the offloading channel. Subsequently, semantic tasks are offloaded to the UAV during the hovering phase. During the process of semantic task offloading, each user can select only one UAV for task offloading, while a single UAV can provide semantic task processing services for multiple users. Additionally, due to the limited channel resources in the environment, each UAV occupies only one channel in a time slot t, and when multiple users choose the same channel to offload tasks to a UAV, they must share that channel. Throughout the UAV-assisted MEC process, each intelligent jammer transmits jamming power towards a specific UAV, interfering with the task offloading process of users to the UAV.

B. Communications Model

Similar to [28], the UAVs in this paper feature a semantic-aware network based on orthogonal frequency-division multiple access (OFDMA). UAVs typically fly at higher altitudes, and communication between UAVs and ground users can be seen as LoS channels. Additionally, due to obstructions from buildings or terrain, some channels become non-line-of-sight (NLoS). In NLoS channels, the communication signal may experience multiple reflections and scattering due to indirect transmission paths, leading to random changes in signal amplitude and phase, exhibiting characteristics of rapid fading, known as Rayleigh fading. This paper's channel model discusses the impacts of both LoS and NLoS channels, where the NLoS channel includes the effects of Rayleigh fading. The expression for the probability of the LoS channel between user m and UAV n at time slot t [28], [40] is given by

$$P_{t,mn}^{LoS} = \frac{1}{1 + \omega_1 \exp\left\{-\omega_2(\theta_{t,mn} - \omega_1)\right\}}, \tag{1}$$

where $\theta_{t,mn}$ represents the elevation angle between user m and UAV n at time slot t, expressed as $\theta_{t,mn} = (180/\pi)\arctan(H/d_{t,mn}^{\mathrm{hor}})$, and $d_{t,mn}^{\mathrm{hor}}$ is the horizontal distance between user m and UAV n at time slot t. ω_1 and ω_2 are coefficients related to the environment [28].

The channel gain at time slot t is given by

$$h_{t,mn} = \frac{10^{-\frac{(\eta_{\mathrm{LoS}} - \eta_{\mathrm{NLoS}})P_{t,mn}^{\mathrm{LoS}} + \eta_{\mathrm{NLoS}}}{10}}}{\left(\frac{4 d_{t,mn}\pi f_c}{c}\right)^2}, \tag{2}$$

where η_{LoS} and η_{NLoS} are the free-space path loss gains corresponding to LoS and NLoS connections, respectively. $d_{t,mn}$ denotes the distance at time slot t.

Moreover, $\delta_{t,mn}$ denotes the association index between user m and UAV n at time slot t. When user m selects UAV n for offload tasks at time slot t, $\delta_{t,mn} = 1$, otherwise $\delta_{t,mn} = 0$. Additionally, $\rho_{t,n}$ is utilized as the channel selection for users who choose UAV n to task offloading at time slot t, the bandwidth allocated to $\rho_{t,n}$ is $B_{t,n}^c$, $B_{t,mn}^c$ refers to the bandwidth allocated to user m who selects UAV n to offload tasks. If user m chooses UAV n to offload tasks,

$B^c_{t,mn} = B^c_{t,n}/\chi_{t,n}$, where $\chi_{t,n}$ refers to the number of users association UAV n at time slot t.

When a user offloads tasks to a UAV, a jammer transmits jamming power towards the UAV. Therefore, at time slot t, when $\tau_{t,n} = \sum_{m=1}^{M} \delta_{t,mn}$ is not equal to 0, the Signal-to-Interference-plus-Noise Ratio (SINR) from user m to UAV n is given by

$$\gamma_{t,mn} = \frac{\delta_{t,mn} h_{t,mn} p_{t,mn} \left| h^{ra}_{t,mn} \right|^2}{h^j_{t,nn} p^{j,m}_{t,nn} \left| h^{j,ra}_{t,nn} \right|^2 + B^c_{t,mn} N_0}, \tag{3}$$

where $\sum_{m=1}^{M} \delta_{t,mn}$ denotes the total number of users using channel $\rho_{t,n}$ for task offloading at time slot t, and if $\sum_{m=1}^{M} \delta_{t,mn} = 0$, then $\gamma_{t,mn} = 0$. $p_{t,mn}$ is the transmission power of user m offloading tasks to UAV n at time slot t, N_0 is the power spectral density. $p^{j,m}_{t,nn}$ refers to the jamming power transmitted by jammer n to UAV n and is distributed to the user m who chooses UAV n for task offloading at time slot t, where $p^{j,m}_{t,nn} = p^j_{t,nn}/\chi_{t,n}$, $p^j_{t,nn}$ denotes the jamming power emitted by jammer n to UAV n, and $h^{ra}_{t,mn} \sim \mathcal{CN}(0,1)$ is the Rayleigh fading coefficient for the sub-channel assigned to user m to UAV n, $h^{j,ra}_{t,nn} \sim \mathcal{CN}(0,1)$ is the Rayleigh fading coefficient for the sub-channel assigned to jamming n to UAV n. $h^j_{t,nn}$ is the channel gain between jammer n and UAV n, expressed as

$$h^j_{t,nn} = \frac{10^{-\frac{(\eta_{\mathrm{LoS}} - \eta_{\mathrm{NLoS}}) P^{j,\mathrm{LoS}}_{t,nn} + \eta_{\mathrm{NLoS}}}{10}}}{\left(\frac{4 d^j_{t,nn} \pi f_c}{c}\right)^2}, \tag{4}$$

where, $P^{j,\mathrm{LoS}}_{t,nn}$ is the LoS channel probability between jammer n and UAV n:

$$P^{j,\mathrm{LoS}}_{t,nn} = \frac{1}{1 + \omega_1 \exp\left\{ -\omega_2 (\theta^{j,\mathrm{LoS}}_{t,nn} - \omega_1) \right\}}, \tag{5}$$

where $\theta^{j,\mathrm{LoS}}_{t,nn}$ is the elevation angle between jammer n and UAV n at time slot t, expressed as $\theta^{j,\mathrm{LoS}}_{t,nn} = (180/\pi) \arctan(H/d^{j,\mathrm{hor}}_{t,nn})$, and $d^{j,\mathrm{hor}}_{t,nn}$ is the horizon distance between jammer n and UAV n at time slot t, $d^j_{t,nn}$ is the distance at time slot t.

Unlike the conventional Shannon formula expressing communication rates, semantic rate is based on semantic information. The semantic rate is defined as the amount of semantic information sent to the transmission medium per second [11] and is measured in $suts/s$. For the pre-trained DeepSC model, the semantic similarity at time slot t from user m to UAV n is defined as $\varpi_{t,mn} = \varpi(K, \gamma_{t,mn})$ [11], where K represents the average number of semantic symbols used for each word in the semantic task offloaded from user m to UAV n at time slot t. The average semantic information for each sentence offloaded from user m to UAV n at time slot t is denoted as ψ^s_t, and the average word length for each sentence is defined as ψ^w_t. The semantic rate at time slot t from user m to UAV n can be expressed as

$$r_{t,mn} = \frac{\delta_{t,mn} B^c_{t,mn} \psi^s_t \varpi_{t,mn}}{\psi^w_t K}, \tag{6}$$

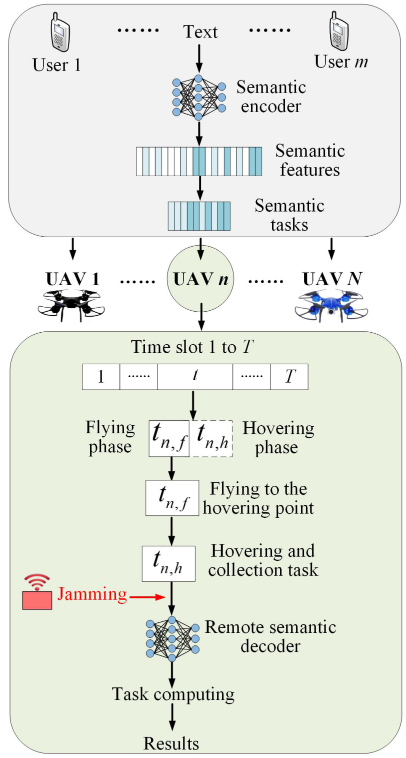


Fig. 2. Semantic task processing model.

where ψ^s_t, ψ^w_t, and K are the parameter dependent on the pre-trained DeepSC model and the source type, and can be treated as a constant during the training process.

In this paper, the quality of semantic communication is assessed using the concept of semantic spectral efficiency (SSE) proposed in [11]. The SSE refers to the rate at which semantic information can be successfully transmitted per unit bandwidth, measured in $suts/s/Hz$. The expression for SSE at time slot t when user m offloads tasks to UAV n is given by

$$\phi_{t,mn} = \frac{r_{t,mn}}{B^c_{t,mn}} = \frac{\delta_{t,mn} \psi^s_t \varpi_{t,mn}}{\psi^w_t \mathrm{K}}, \tag{7}$$

where $B^c_{t,mn}$ refers to the bandwidth allocated to user m who selects UAV n to offloading task. The total sum of SSE within one time slot t is given by

$$\phi_t = \sum_{m=1}^{M} \sum_{n=1}^{N} \phi_{t,mn}. \tag{8}$$

The total sum of SSE within one episode is SSEs, expressed as

$$\phi = \sum_{t=1}^{T} \phi_t. \tag{9}$$

When the value of ϕ is larger, it indicates that a greater semantic rate can be transmitted per unit of bandwidth, reducing the time required to transmit tasks of the same size, thereby enhancing the performance of semantic communication.

C. Task Completion Time Model

At time slot t, UAV n first flies for a duration $t_{f,n}$, and then hovers at a suitable point to receive tasks, with a hovering time $t_{h,n} = T_s - t_{f,n}$. The semantic task volume received by n-th UAV during this time is denoted as $c_{t,n}^{se} = \sum_{m=1}^{M} \min(r_{t,mn}t_{h,n}, \delta_{t,mn}s_{t,m}^{d}\psi_t^s)$ [11].

The communication delay incurred by UAV n during this period is denoted as

$$D_{t,n}^{com} = \sum\nolimits_{m=1}^{M} \min(t_{h,n}, \frac{\delta_{t,mn}s_{t,m}^{d}\psi_t^s}{r_{t,mn}}). \tag{10}$$

The total communication delay is the sum of the delays of each UAV, the total communication delay incurred by all N UAVs at time slot t is expressed as

$$D_t^{com} = \sum\nolimits_{n=1}^{N} D_{t,n}^{com}. \tag{11}$$

Similar to [11], after the semantic task is transmitted to the UAV and decoded, the actual task volume received by UAV n is given by $c_{t,n} = c_{t,n}^{se}s_t^v/\psi_t^s$. Subsequently, these tasks are computed on the UAV, and the computation delay is given by

$$D_t^{rc} = \sum\nolimits_{n=1}^{N} \frac{c_{t,n}s_t^l}{f_n^u s_t^v}, \tag{12}$$

where f_n^u represents the computing capability of the UAV n.

Therefore, the total task completion time at time slot t is

$$D_t = D_t^{com} + D_t^{rc}. \tag{13}$$

The algorithm in our work integrates semantic communication with the allocation of computing and communication resources. The smaller the task completion time, the shorter the task transmission and computation time, indicating that the algorithm's strategy for allocating computing and communication resources is more effective.

D. Problem Formulation

The objective of this paper is to minimize the task completion time by optimizing the strategies of UAV movement trajectory $\mathbf{L}$, user association $\boldsymbol{\delta}$, and user channel selection $\boldsymbol{\rho}$, respectively, formulated as

$$\begin{aligned} \mathbf{P1:} \max_{\mathbf{L},\boldsymbol{\delta},\boldsymbol{\rho}} & \sum\nolimits_{t=1}^{T} -(\lambda_1 D_t^{com} + \lambda_2 D_t^{rc}) \\ s.t.\ & (\text{C1}) : x_{\min} \leq l_x \leq x_{\max}, \\ & (\text{C2}) : y_{\min} \leq l_y \leq y_{\max}, \\ & (\text{C3}) : \varpi_{th} < \varpi_{t,mn}, \\ & (\text{C4}) : 1 \leq c \leq C, \\ & (\text{C5}) : \sum\nolimits_{n=1}^{N} c_{t,n} = c_{t,sum}, \end{aligned} \tag{14}$$

where λ_1 and λ_2 refer to the task communication time coefficient and the task computation coefficient, respectively. The primary function of λ_1 and λ_2 is to adjust the magnitudes of D_t^{com} and D_t^{rc} to make them generally equal, thus aligning their importance during the optimization process. When UAVs with abundant computing resources are far from users, the communication delay required for users to offload tasks is longer, but the computation delay is shorter when offloaded to these resource-rich UAVs. The system needs to balance between D_t^{com} and D_t^{rc} to achieve an better optimization strategy. (C1) and (C2) represent to the UAV flight range limitations, and this paper specifies that the UAV moves within the specified range. (C3) refers to the minimum semantic similarity threshold. Specify that the semantic similarity should be greater than the minimum semantic similarity threshold ϖ_{th}. (C4) refers to the channel number limitation, indicating that there is a limited number of channels in the environment. (C5) refers to the task limitation, indicates that the UAV should receive all tasks generated by the user at each time slot.

III. Problem Transformation

The optimization problem described in Eq. (14) is notably challenging, as it is both non-convex and falls into the category of NP-hard problems. Consequently, obtaining an optimal solution in an efficient manner proves to be a daunting task. Moreover, in a real-world UAV-assisted MEC network with semantic communication, dynamic factors such as UAV trajectories, users' task volumes, channel conditions, jamming patterns, and strategies for semantic task offloading are constantly changing. Traditional algorithms require less data and do not need to train models using large-scale data, resulting in lower computational requirements and higher computational efficiency. However, they are unable to interact with and learn from dynamic environments to effectively adapt to them. Therefore, we employ a DRL algorithm to tackle the problem. DRL algorithms offer adaptability to evolving environments and tasks, making them well-suited for addressing challenges in dynamic settings. Furthermore, they possess exploratory capabilities that enable them to experiment with new strategies independently, thereby discovering optimal solutions to complex problems.

A. DRL Strategies for Users and UAVs

We model the large-scale access decision problem by transforming the optimization problem **P1** into a Markov decision process (MDP) within the DRL framework. DRL can be defined using the concept of MDP, denoted as $\langle U^{us}, S^{us}, A^{us}, P^{us}, r^{us}\rangle$ and $\langle U^{uav}, S^{uav}, A^{uav}, P^{uav}, r^{uav}\rangle$, respectively. Here, U^{us} denotes the set of users as agents, S^{us} denotes the set of user states, A^{us} denotes the set of user actions, P^{us} denotes the probability of state transitions for users, and r^{us} denotes the rewards for users. U^{uav} denotes the set of UAVs as agents, S^{uav} denotes the set of UAV states, A^{uav} denotes the set of UAV actions, P^{uav} denotes the probability of state transitions for UAVs, and r^{uav} denotes the rewards for UAVs.

During the execution of tasks by the UAV, it is assumed that the position of the jammer can be observed. During this process, the distance between the UAV and the jammer is estimated through learning. The impact of the jammer's jamming power and other factors on the system is mainly reflected in the reward. These jamming factors are not directly observable and can only be known through accumulating experience, thus there are estimation errors and delays in the system's recognition of these unobservable factors. These errors and delays are incorporated into training along with accumulated experience. Through multiple training, more accurate jamming information can be obtained. Subsequently, when encountering similar situations with estimation errors and delays, the system can quickly identify and accurately determine the jamming information, while swiftly adjusting UAV trajectories, user associations, and channel selections to adapt to the jamming environment. By ensuring the performance of UAV-assisted MEC while staying as far away from jammers as possible, and by reasonably allocating computing and communication resources to counteract jamming attacks, the system enhances its robustness to jamming attacks and estimation errors. This results in a better strategy for UAV-assisted MEC under jamming attacks. Therefore, the decision problem in this paper can be represented as an MDP problem constructed as follows.

1) Agent U^{us}: In the system, the intelligent agents are M ground users.

Translate the information available from optimization problem **P1**, such as the positions of the UAV, users, and jammer, as well as channel selection, into the user's DRL state.

2) State Space S^{us}: In this paper, it is assumed that the UAV can observe the positions of the UAV itself, the users, and the jammer, as well as the channel selection status when the UAV receives tasks. The jamming power of the jammer is determined through the accumulation of experience, hence this information is somewhat vague and subject to delays. However, after a period of training, more accurate values can be obtained. We define the vector $s_t^{us} = \left\{ l_{t,m}^{us}, l_{t,n}^{uav}, l_{t,n}^{j}, \rho_{t-1,n} \right\}$ as the state space of user at time slot t, where $l_{t,m}^{us}$ refers to the position of user m at time slot t, and $l_{t,n}^{uav}$ refers to the position of UAV n at time slot t, $l_{t,n}^{j}$ refers to the position of jammer n at time slot t, $\rho_{t-1,n}$ refers to the channel selected by the user who selects the UAV n for task offloading at time slot $t-1$, $m \in \{1, 2, \cdots, M\}$, $n \in \{1, 2, \cdots, N\}$.

Translate the optimization variables related to users in optimization problem **P1**, such as user associations and channel selections, into actions for user's DRL.

3) Action Space A^{us}: We define $a_t^{us} = \left\{ \delta_{t,mn}, \rho_{t,n} \right\}$ as the action space of user at time slot t, where $\delta_{t,mn}$ refers to the user associations coefficient between user m and UAV n at time slot t, and $\rho_{t,n}$ refers to the channel selected by the user who selects the UAV n to offloading the task at time slot t, $m \in \{1, 2, \cdots, M\}$, $n \in \{1, 2, \cdots, N\}$. When the jammer interferes with the system, users plan user associations and channel selections for task offloading, cooperating with the UAV to efficiently execute tasks while minimizing the impact of jamming attacks.

4) Transition Probability P^{us}: The transition probability P^{us} of a user refers to the probability that the user transfers from a state s_t^{us} to the next state s_{t+1}^{us} after performing an action a_t^{us}.

Translate the optimization objectives and constraints from optimization problem **P1** into rewards for user's DRL.

5) Reward r^{us}: The user's reward is the same as the system reward, $r_t^{us} = r_t = -(\lambda_1 D_t^{com} + \lambda_2 D_t^{rc}) - \lambda_3 - \lambda_4 - \lambda_5$, λ_3 refers to the penalty imposed when the set minimum semantic similarity threshold ϖ_{th} is not met, $\lambda_3 = \lambda_3^1 \lambda_t$, and λ_3^1 refers to a single penalty coefficient, λ_t refers to the number of users whose communication link has not reached the minimum semantic similarity threshold at time slot t, λ_4 refers to the penalty for not receiving all user tasks, λ_5 refers to the penalty coefficient for collision between UAVs. Users can estimate the impact of the jammer on the system based on the reward, which requires some time to accumulate experience, thus it is somewhat vague and delayed.

The details of the UAV's MDP model are as follows.

6) Agent U^{uav}: In the system, N UAVs participate in the MEC, and the agent is N UAVs.

Translate the information available from optimization problem **P1**, such as the positions of the UAVs, users, and jammer, as well as channel selection, into the UAV's DRL state.

7) State Space S^{uav}: Like the users, UAVs can observe the positions of the UAVs, users, and jammer, as well as the channel selection status when receiving tasks. Similarly, the jamming power can be determined through the accumulation of experience, thus the size of the jamming power is somewhat vague and delayed. After a period of training, more accurate values of the jamming power can be obtained. We define the vector $s_t^{uav} = \left\{ l_{t,m}^{us}, l_{t,n}^{uav}, l_{t,n}^{j}, \rho_{t-1,n} \right\}$ as the state space of UAVs at time slot t, where $l_{t,m}^{us}$ refers to the position of user m at time slot t, $l_{t,n}^{uav}$ refers to the position of UAV n at time slot t, $l_{t,n}^{j}$ refers to the position of jammer n at time slot t, and $\rho_{t-1,n}$ refers to the channel selected by the user who chooses UAV n for task offloading at time slot $t-1$. $m \in \{1, 2, \cdots, M\}$, $n \in \{1, 2, \cdots, N\}$.

Translate the optimization variables related to UAVs in optimization problem **P1**, such as the UAV's movement trajectory, into actions for UAV's DRL.

8) Action Space A^{uav}: We define $a_t^{uav} = \left\{ d_{t,n}^{uav}, \theta_{t,n}^{uav} \right\}$ as the action space of UAV at time slot t, where $d_{t,n}^{uav}$ refers to the distance moved by UAV n at time slot t, and $\theta_{t,n}^{uav}$ refers to the direction moved by UAV n at time slot t. $n \in \{1, 2, \cdots, N\}$. When the jammer transmits jamming power, the UAV plans its movement trajectory to find a suitable hovering point. By cooperating with users, the system can minimize the impact of jamming attacks while also ensuring a higher semantic rate to efficiently execute tasks.

9) Transition Probability P^{uav}: The transition probability P^{uav} of a UAV refers to the probability that the UAV transitions from one state s_t^{uav} to the next state s_{t+1}^{uav} after executing an action a_t^{uav}.

Translate the optimization objectives and constraints from optimization problem **P1** into rewards for UAV's DRL.

10) Reward r^{uav}: The reward for a UAV is consistent with the system reward, $r_t^{uav} = r_t = -(\lambda_1 D_t^{com} + \lambda_2 D_t^{rc}) - \lambda_3 - \lambda_4 - \lambda_5$. Like the users, the UAV also estimates the impact of the jammer on the system based on the reward, which requires a period of experience accumulation, thus it is somewhat vague and delayed.

In the system, the perception of jamming information is relatively vague and may involve certain estimation errors and delays. These errors and delays are significantly reduced to negligible levels through training and learning, after accumulating a large amount of experience in the initial stages.

B. DRL Strategies for Intelligent Jammers

During the operation of the jammer, it observes the position of UAVs and the channel selection status of users.

By observing and analyzing over several time slots, the jammer predicts the operational patterns of UAVs and users, accumulates relevant experience, and gathers information to disrupt the operation of both UAVs and users. The process of UAV-assisted MEC under a jamming environment can be viewed as a game among UAVs, users, and the jammer, similar to the process described in the literature [41]. The UAV disrupts MEC by emitting jamming power. Initially, this reduces the system's reward, but subsequently, UAVs and users gradually increase their gains by optimizing UAV trajectories, user associations, and channel selections strategies. Eventually, a balance is achieved between the execution of system tasks and the jamming process, leading to the convergence of the reward curve. In this paper, we consider that the jammer can observe the position of the UAV and emit jamming power to the UAV during the UAV's task, with the goal is to increase the task completion time of system tasks. Therefore, the optimization problem of the smart jammer is formulated as

$$\textbf{P2:} \max(\sum_{t=1}^{T} D_t). \tag{15}$$

We model the large-scale access decision problem by transforming optimization problem **P2** into a markov decision process (MDP) within the DRL framework. DRL can be defined using the concept of MDP, denotes $\langle U_j, S_j, A_j, P_j, r_j \rangle$. Where U_j denotes the set of agents acting as jammers, S_j denotes the set of states for the jammer network, A_j denotes the set of actions for the jammers, P_j denotes the transition probability for the jammer's states, and r_j denotes the reward for the jammers. Therefore, the DRL method can quickly derive the optimal strategy. When UAVs and users are executing tasks, the position of the UAVs can be directly observed by the jammers, while the channel selection for users offloading tasks to the UAVs can only be vaguely perceived at the beginning. After extensive training, by summarizing experience and combining it with the jammers' rewards, more precise channel selection can be achieved. Thus, the state of the jammer is somewhat vague and has latency.

1) Agent U^j: In the system, N jammers participate in the jamming process of jamming UAVs, and the agent is N jammers.

Translate the information available from optimization problem **P2**, such as the positions of the UAV and jammer, as well as channel selection, into the jammer's DRL state.

2) State Space S^j: We define the vector $s_t^j = \left\{ l_{t,n}^{uav}, l_{t,n}^j, \rho_{t-1,n} \right\}$ as the state space of the jammer at time slot t, where $l_{t,n}^{uav}$ refers to the position of UAV n at time slot t, $n \in \{1, 2, \cdots, N\}$, $l_{t,n}^j$ refers to the position of jammer n at time slot t, $\rho_{t-1,n}$ refers to the channel selected by the user who selects the UAV n to offloading the task at time slot $t-1$, $n \in \{1, 2, \cdots, N\}$. The jammer observes the position of the UAV and gains experience about the channel selection when users offload tasks to the UAV. This state is somewhat vague and delayed, but a clearer state can be achieved after a period of training and experience accumulation.

Translate the optimization variables related to the jammer in optimization problem **P2**, such as the jammer's transmission power, into actions for the jammer's DRL.

3) Action Space A^j: We define $a_t^j = \left\{ p_{t,nn}^j \right\}$ as the action space of the jammer at time slot t, where $p_{t,nn}^j$ refers to the jamming power emitted by the jammer n in the direction of the UAV n at time slot t, $n \in \{1, 2, \cdots, N\}$.

4) Transition Probability P^j: The transition probability of a jammer refers to the probability P^j that the jammer transfers from one state s_t^j to the next s_{t+1}^j after executing an action a_t^j.

Translate the optimization objectives and constraints from optimization problem **P2** into rewards for the jammer's DRL.

5) Reward r^j: the reward function is used to evaluate the learning strategy, we consider that the jammer can be informed of the rewards of the UAV and the user system, and the jammer reward is the negative of the system reward, $r_t^j = -r_t - \lambda_6$, λ_6 refers to a penalty that is proportional to the power emitted by the jammer, because the jammer itself has a limited power. Therefore, higher jamming power is not always better. The jammer can gradually obtain a clear state of jamming by combining the rewards received with the experience gained from multiple training sessions.

The jammer learns the state-action tuple $\left\langle U^j, S^j, A^j, P^j, r^j \right\rangle$ in conjunction with DRL to derive a policy that minimizes the system rewards in order to jam the system.

In real-world scenarios, the state transition probabilities P are often unattainable, rendering traditional MDP-based decision-making approaches impractical. To overcome this challenge, we introduce a model-free RL method to address decision-making under uncertain state transitions. This dynamic planning technique enables the derivation of an optimal strategy by iteratively engaging with the environment, all without the need for constructing a predefined environment model. Consequently, model-free RL methods are well-suited for tackling complex challenges. We leverage the model-free RL methods to optimize the strategies, considering the complex real-world dynamics.

IV. UAV-Assisted MEC Scheme Combining T5D and Semantic Communication

DRL and semantic communication can be effectively integrated to enhance the performance of UAV-assisted MEC. Performing UAV-assisted MEC in a jamming environment leads to a reduction in SINR during communication, which in turn decreases semantic similarity and semantic rate. Reinforcement learning can improve SINR by adjusting UAV-assisted MEC strategies, thereby enhancing the semantic rate and increasing the reward of UAV-assisted MEC. In this section, we discuss relevant DRL algorithms for this paper, such as Deep Q-Network (DQN), Double Deep Q-Network (DDQN), Deep Deterministic Policy Gradient (DDPG), and Twin Delayed Deep Deterministic Policy Gradient (TD3). Subsequently, we introduce the framework and algorithmic workflow of the T5D DRL algorithm proposed in this paper. Finally, we delve into the cooperative and adversarial strategies among users, UAVs, and jammers within the context of this paper.

In recent years, due to the increasing complexity of communication environments, traditional algorithms such as gradient descent and greedy algorithm have struggled to effectively address issues arising in dynamic communication scenarios. RL algorithms like Q-learning have emerged as viable solutions. In contrast to traditional algorithms, the RL algorithms employ a Q-table to store Q-values for state-action pairs and possess autonomous learning capabilities. They interact with the environment, adapt, and self-improve their strategies. Moreover, they have the capacity for exploration, allowing them to discover potentially high-reward strategies. However, Q-learning can only handle problems with limited state and action spaces. In increasingly complex environments with vast state and action spaces, Q-learning is inadequate for finding optimal strategies. This is where DRL algorithms, such as DQN and DDQN, which integrate neural networks, come to the fore.

A. DDQN Algorithm

The DDQN algorithm stands apart from traditional RL algorithms. Unlike standard approaches, the DQN algorithm leverages a deep neural network to approximate the Q-function. It takes states as inputs and produces Q-values for all possible actions. This unique capability equips the DQN algorithm to efficiently tackle problems characterized by extensive state-action spaces. However, during the learning process, the DQN algorithm can tend to overestimate Q-values. To address this overestimation challenge, researchers introduced the DDQN algorithm, an enhanced version of the DQN algorithm. The DDQN algorithm employs two neural networks for Q-value estimationłone for optimal action selection and the other for target Q-value estimation. This dual-network approach helps mitigate overestimation errors. Furthermore, the DDQN algorithm incorporates an experience pool, where computational experiences are stored. The network parameters are continually updated using a loss function until the optimal strategy is achieved.

B. DDPG Algorithm

Although the DDQN algorithm has good performance, it is most effective when handling problems with discrete action spaces. When confronted with tasks involving continuous action spaces, the DDQN algorithm struggles to efficiently derive a policy. This limitation has led to the adoption of the DDPG algorithm, particularly for scenarios requiring high-dimensional, precise actions. The DDPG algorithm excels in continuous action spaces and is capable of handling tasks that demand high accuracy. It relies on four neural networks: the actor network, critic network, target actor network, and target critic network. The process begins with the actor network selecting an action based on the current state. The resulting experience is stored in an experience pool, from which a small batch of experiences is extracted. Subsequently, the Q-value of the next state is estimated using both the target actor and target critic networks. During this process, the critic network's loss is computed, and networks are updated. The loss functions employed for updating the actor and critic networks are as follows:

$$L^{DDPG}(\theta^A) = -E\left[Q(s, \mu(s|\theta^A)|\theta^C)\right], \tag{16}$$

$$L^{DDPG}(\theta^C) = E\left[\left(Q(s, a(\theta^C)) - y\right)^2\right], \tag{17}$$

where θ^A is a parameter in the actor network, $\mu(s|\theta^A)$ is the output of the actor network. θ^C is a parameter in the critic network, $a(\theta^C)$ is the output of the critic network, and y refers to the target Q-value.

C. T5D Algorithm

The proposed UAV-assisted MEC scheme, which integrates path planning and resource allocation in a jamming environment, employs the TD3-DDQN algorithm, as shown in Fig. 3. In the TD3 algorithm, an action is initially selected based on the current state using the actor network. Subsequently, the experience is stored in the experience pool, and the Q-value of the next state is estimated using a target critic network. The critic's loss is then computed while updating the critic network. It's important to note that TD3 utilizes two critic networks, labeled Q_1 and Q_2. Following this, the policy loss of the Actor network is calculated and used to update the actor network. Finally, the target network is updated. The loss function is as follows.

$$L^{TD3}(\theta^A) = -E\left[Q_1(s, \mu(s|\theta^A)|\theta^{Q_1})\right], \tag{18}$$

where θ^A is a parameter in the actor network and $\mu(s|\theta^A)$ is the output of the actor network.

The loss functions of the two critic networks are:

$$L^{TD3}(\theta^{Q_1}) = E\left[\left(Q_1(s, a(\theta^{Q_1})) - y\right)^2\right], \tag{19}$$

$$L^{TD3}(\theta^{Q_2}) = E\left[\left(Q_2(s, a(\theta^{Q_2})) - y\right)^2\right], \tag{20}$$

θ^{Q_1} is the parameter of Q_1, $a(\theta^{Q_1})$ refers to the output of Q_1. θ^{Q_2} is the parameter of Q_2, $a(\theta^{Q_2})$ refers to the output of Q_2, y refers to the target Q-value. The process of UAV-assisted MEC scheme combining T5D and semantic communication algorithm mentioned in this article is shown in **Algorithm 1**.

Remark 1: In the initial phase, each user transforms their tasks into semantic tasks through the DeepSC model. Subsequently, they select a UAV associations, and choose the task offloading channel. UAVs receive the tasks from users through a combination of flying and hovering at predefined hovering points. Simultaneously, an equal number of jammers in the system matches the number of UAVs, with each jammer continuously transmitting jamming power directed at its respective UAV. As task processing time progresses, both users and UAVs accumulate experiences, which enable them to learn and formulate optimal strategies for UAV flight trajectories, user associations, and offload channel selections. These strategies aim to maximize the system's rewards. Conversely, jammers also accumulate experience, shaping their strategy to disrupt user task offloading to UAVs by selecting optimal jamming power levels. Their goal is to minimize the system's rewards. When the jammer transmits jamming power towards the UAV,

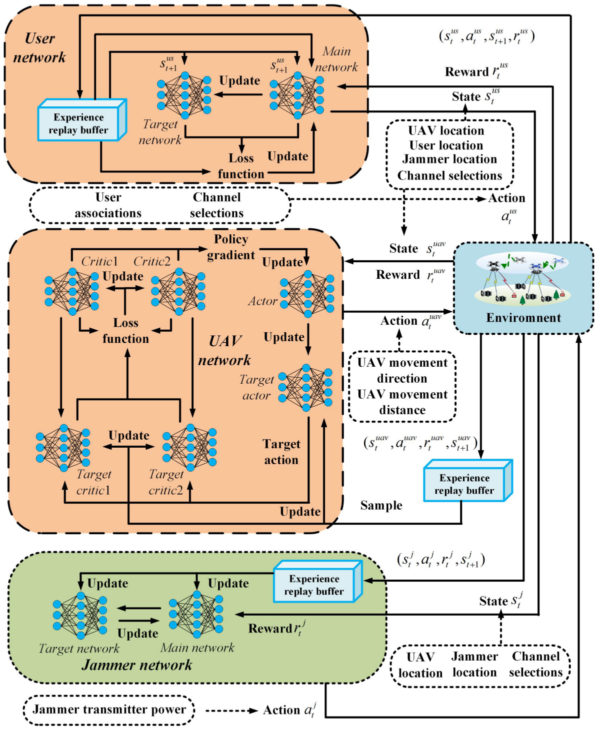


Fig. 3. Network diagram of the T5D algorithm.

the UAV optimizes its movement trajectory to minimize the impact of jamming attacks while ensuring the performance of the system while keeping the UAV as far away from the jammer as possible. When users become aware of jamming attacks on the communication link, they cooperate with the UAV to optimize user associations strategies and channel selections while the UAV optimizes its trajectory. Through extensive training, they strategically plan user associations and channel allocation to minimize the impact of jamming attacks on the system, ensuring a good semantic rate. Therefore, the UAV and users jointly optimize the UAV trajectories, user associations, and channel selections to resist jamming attacks and to optimize UAV-assisted MEC strategies. These strategies aim to maximize the system's rewards. The strategies of the users and UAVs reach a dynamic equilibrium with the jammer's strategy at a certain point in time, and the reward then converges. Since it is impossible to perceive all the information in the dynamic environment, only the most important information from the dynamic environment is selected for learning. Therefore, although the convergence value is not the optimal solution, it is the best solution that can be achieved.

Remark 2: UAVs and users can share relevant state information among themselves and between each other, including the jamming information each has perceived. When the communication link between a user and a UAV is attacked by a jammer, the two UAVs cooperate strategically with each user. Under the shared jamming information, each UAV reasonably allocates its movement trajectory and hovering points, plans channel selection to allocate communication resources, and associates non-redundantly with a subset of users based on its own computing resources. It helps receive user offloading tasks, reduce the impact of jamming attacks on the system, and maintain the stability of system resource management.

Algorithm 1 UAV-Assisted MEC Scheme Combining T5D and Semantic Communication Algorithm

Initialize: Parameters of the networks of UAVs, users, and jammers.
for *episode* = 1,…,1000 **do**.
  Initialize the information of UAV's position, user's position, jammer's position etc. to get the initial state of UAV s_1^{uav}, user s_1^{us}, jammer s_1^{j}. It also randomly assigns initial values such as user associations $\delta_{1,mn}$, user channel selections $\rho_{1,mn}$.
  for t = 1,2,…,T **do**.
    The user decision network selects the user action a_t^{us} according to the state s_t^{us}.
    The UAV decision network selects a UAV action a_t^{uav} based on the state s_t^{uav}.
    The jammer decision network selects the jammer action a_t^{j} according to the state s_t^{j}.
    Each action is applied to the environment according to the relevant equations in Eq. (1)-(13).
    The corresponding reward r^{uav}, r^{us}, r^{j} is obtained.
    The environment is changed due to the action and the next environment s_{t+1}^{uav}, s_{t+1}^{us}, s_{t+1}^{j} is obtained.
    Store $(s_t^{uav}, a_t^{uav}, r_t^{uav}, s_{t+1}^{uav})$ into the experience pool.
    Randomly select a part of the experience pool to update the UAV network parameters.
    Store $(s_t^{us}, a_t^{us}, r_t^{us}, s_{t+1}^{us})$ into the experience pool.
    if $t_1 > t_{th}^{us}$, every t_{th}^{us} steps randomly select a part of the experience pool to update the user network parameters.
    end if.
    Save $(s_t^{j}, a_t^{j}, r_t^{j}, s_{t+1}^{j})$ to the experience pool.
    if $t_1 > t_{th}^{j}$, every t_{th}^{j} steps randomly select a portion of the experience pool to update the jammer network parameters.
    end if.
    $t_1 = t_1 + 1$.
  end for.
end for.
Obtain the strategy.

1) Computational Complexity Analysis: This section provides a complexity analysis of the T5D algorithm proposed in this paper. In the TD3 part of the algorithm, J and V are defined as the number of DNN layers in the actor network and the critic network, respectively. J_j is defined as the number of neurons in the j-th layer of the actor network, and V_i is defined as the number of neurons in the i-th layer of the critic network. In the actor network, the computational complexity of the i-th layer is $O(J_{j-1}J_j + J_jJ_{j+1})$, and the total computational complexity is $O(\sum_{j=2}^{J-1}(J_{j-1}J_j + J_jJ_{j+1}))$. Similarly, in the critic network, the total computational complexity is $O(\sum_{i=2}^{I-1}(V_{i-1}V_i + V_iV_{i+1}))$. The TD3 algorithm involves both the actor and critic networks, with the complexity of each training iteration being $O(\sum_{j=2}^{J-1}(J_{j-1}J_j + J_jJ_{j+1}) + \sum_{i=2}^{V-1}(V_{i-1}V_i + V_iV_{i+1}))$.

In this paper, there are E episodes, each with T time slots, so the computational complexity O_1 during training is $O(ET(\sum_{j=2}^{J-1}(J_{j-1}J_j + J_jJ_{j+1}) + \sum_{i=2}^{V-1}(V_{i-1}V_i + V_iV_{i+1})))$. In the DDQN part of the algorithm, L, B_0, and B_l respectively represent the number of training layers proportional to the number of states, the size of the input layer, and The number of neurons used in the l-th layer of DDQN. At each step, the computational complexity for each agent is $O(B_0B_1 + \sum_{l=1}^{L-1} B_lB_{l+1})$, and during the training process, with E episodes, T time slots, and T_I iterations, the total computational complexity O_2 is $O(ETT_I(B_0B_1 + \sum_{l=2}^{L-1} B_lB_{l+1}))$.

The complexity of the algorithm presented in this paper is the sum of the complexities of two algorithms. Due to the use of neural networks, its complexity far exceeds that of traditional algorithms and reinforcement learning algorithms that do not use neural networks, such as the Q-learning algorithm. At the same time, the complexity of algorithm in our work is larger than that of DRL algorithms that use only continuous actions like TD3 and DDPG, or those that use only discrete actions like DDQN and DQN. Although the complexity of this paper's algorithm is large, the paper primarily optimizes the UAV-assisted MEC strategy under jamming attacks by adjusting UAV trajectories, users associations, and channel selections. UAVs trajectories involve continuous actions, while user associations and channel selections involve discrete actions, requiring the use of different types of DRL algorithms for different types of actions to enhance decision-making performance. Moreover, in the T5D algorithm, we have streamlined the neural networks of the TD3 and DDQN algorithms while ensuring performance, resulting in a simplified and efficient algorithm for UAV-assisted MEC under jamming attacks. Therefore, although the algorithm presented in this paper is complex, its main purpose is to enhance system decision-making performance, and it has been streamlined to ensure that the complexity remains within an acceptable range.

Theorem: In the network part of the T5D algorithm, convergence can be achieved using policy evaluation and policy enhancement within the network, meaning that $\pi(a_t|s_t;\theta^A)$ converges to policy $\pi^*(a_t|s_t;\theta^A)$. At the same time, the following four conditions must be satisfied [42], [43]: 1) The learning rate $\alpha_{t,l}$ of the network at time slot t should be: $\sum_{t=0}^{\infty} \alpha_{t,l} = \infty$, and $\sum_{t=0}^{\infty} (\alpha_{t,l})^2 = \infty$. 2) The instantaneous reward r_t is bounded. 3) The policy function $\pi(a_t|s_t;\theta^A)$ is continuously differentiable within θ^A. 4) The sequence (s_t, a_t, r_t) is independent and identically distributed.

V. Simulation Results and Discussion

In this paper, there are two UAVs and ten ground users. The number of sentences for each user follows a Poisson distribution with a parameter of 1000, and the average size of each sentence is 1200 bits. There are two available channels in the environment with channel bandwidths of 4MHz and 2MHz, respectively. The user's transmission power is fixed at 0.5W, and the range of jammer's transmission power $p_{t,nn}^j$ is $\{14, 16, 18, 20, 22\}$ dBm. Consistent with the literature [28], each round in this paper has a total time of 600s, with K=4. λ_1= 0.066, λ_2= 0.033, λ_3= $-0.3\lambda_t$, λ_t refers to the number of users whose communication link has not reached the minimum semantic similarity threshold at time slot t.

In T5D, the UAVs decision-making algorithm is TD3, with an actor learning rate of 0.00001, a critic learning rate of 0.001, two hidden layers each with 256 neurons, a memory pool capacity of 3000, a batch size of 100, a replay buffer size of 1600, and a discounted factor of 0.9. The user's decision-making algorithm is DDQN, with a learning rate of 0.01, three hidden layers each with 128 neurons, a memory size of 10000, a batch size of 300, and a target network replacement frequency of 200. In DPD, the UAVs decision-making algorithm is DDPG, with all parameters the same as those in T5D or [28]. The jammer's decision-making algorithm is DQN, with a learning rate of 0.001, each hidden layer containing 128 neurons, a memory pool capacity of 10000, and a batch size of 300.

This work compares the performance of six algorithms: T5D, DPD [28], DDQN [30], T5DF1, T5DF2, and NT5D. **T5D:** The TD3 algorithm combined with the DDQN algorithm for resource management in our work. **DPD:** The DDPG+DQN algorithm, where the parameters are consistent with the literature [28]. **DDQN:** The parameters are consistent with the literature [30], which discretizes the actions of UAVs. **T5DF1:** The resource management strategy of half of the UAVs and users are fix in the system. **T5DF2:** The resource management strategy of all UAVs and users are fix in the system. **NT5D:** The algorithm does not incorporate semantic communication for MEC.

Fig. 4 illustrates the cumulative reward curves for five algorithms in the proposed environment. The reward curves of the five algorithms converge to a certain level after a number of episodes, but the reward at convergence of the algorithm proposed is higher than the comparison algorithms, achieving the highest reward at convergence. This is because the system's reward is composed of factors such as task completion time and SSEs, where a smaller task completion time and larger SSEs yield a higher reward. The T5D algorithm proposed in this paper optimizes UAV trajectories, user associations, and channel selections effectively under complex dynamic jamming attacks, increasing the semantic rate during the task offloading process, thereby reducing task completion time and increasing SSEs. Compared to the benchmark algorithms, the T5D algorithm is able to make better decisions because both the UAV network and the user network components of the T5D algorithm consist of dual networks, and the UAV network component also employs a policy delay update mechanism, which reduces the overestimation issue and enhances stability. Furthermore, the T5D algorithm can optimize both discrete and continuous actions. The DPD algorithm suffers from overestimation problems and has lower stability, the DDQN algorithm can only optimize continuous actions, and the UAV trajectories in the T5DF1 and T5DF2 algorithms are restricted, hence they cannot effectively optimize UAV trajectories, user associations, and channel selections simultaneously to counter jamming attacks. Therefore, the T5D algorithm proposed in this paper achieves the optimal reward under jamming attacks

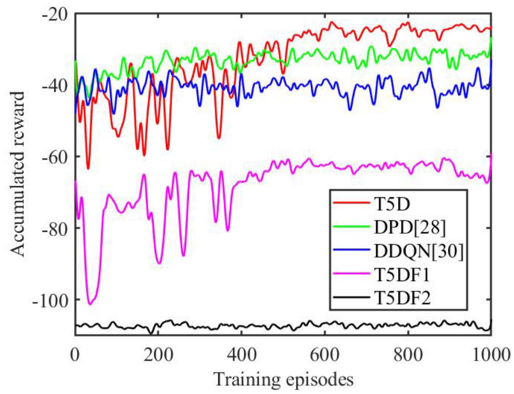
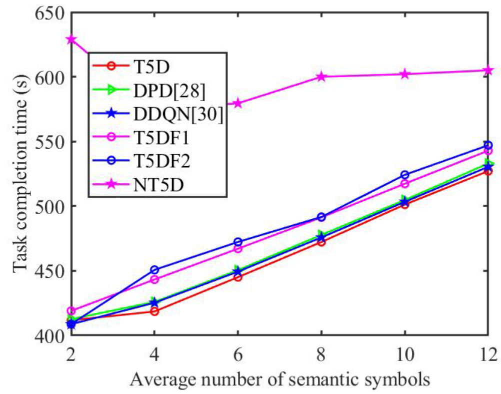


Fig. 4. Reward curves for different algorithms.

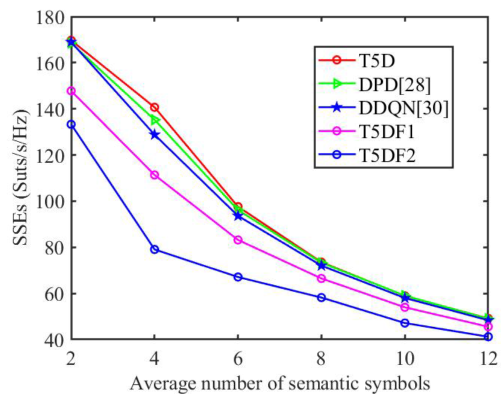
and is the best choice for optimizing UAV-assisted MEC strategies.

Fig. 5 illustrates the comparison of task completion time and SSEs under different average numbers of semantic symbols used for each word K for various algorithms. As depicted in Fig. 5(a), the overall task completion time of different algorithms increases with the rise in K. This is because, as known from Eq. (6), an increase in the value of K causes the semantic rate to decrease. Although a higher K value also results in increased semantic accuracy, the magnitude of increase is smaller, and overall, an increase in K leads to a decrease in the semantic rate. Consequently, as K increases, task completion time increases, and SSEs decrease. The T5D algorithm proposed in this paper effectively reduces Q-value overestimation, enhances algorithm stability, and optimizes both continuous and discrete actions. When the K value changes and the semantic rate varies, the T5D algorithm can adjust UAV trajectories, user associations, and channel selections to adapt to environmental changes and achieve the optimal UAV-assisted MEC strategy. The DPD algorithm suffers from overestimation issues, the DDQN algorithm cannot optimize both continuous and discrete actions simultaneously, and the T5DF1 and T5DF2 algorithms cannot effectively optimize UAV trajectories. Therefore, except when $K{=}2$, the T5D algorithm achieves the best task completion time and SSEs compared to the benchmark algorithms. When $K{=}2$, the SSEs of the T5D algorithm are slightly less than those of the DPD algorithm, but situations where $K{=}2$ are rare. Thus, as K changes, the T5D algorithm continues to achieve the optimal UAV-assisted MEC strategy.

Fig. 6 displays a comparison of task completion time and SSEs under different user quantities for various algorithms. As shown in Fig. 6(a), the task completion time increases with an increasing number of users. This increase is attributed to the larger task volume resulting from a greater number of users. With the same semantic rate, more users lead to greater task completion times and higher accumulated SSEs. Due to the different locations of users, it is also necessary to jointly optimize UAV trajectories, user associations, and channel selections strategies when adding new users to achieve optimal task completion times. From Fig. 6, it is evident that the T5D algorithm achieves the smallest task completion time and the largest SSEs across varying numbers of users. This



(a) Comparison of task completion time of different algorithms

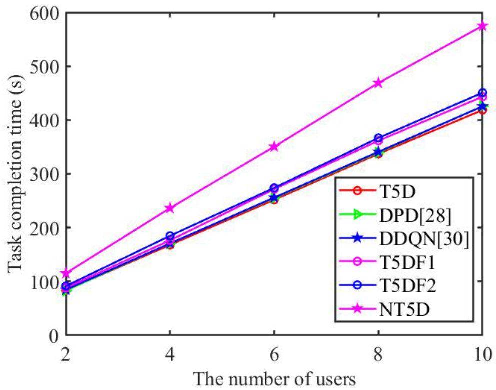


(b) Comparison of SSEs of different algorithms

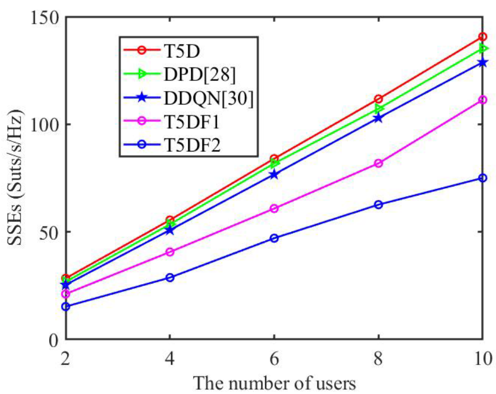
Fig. 5. Performance comparison of different average number of semantic symbols.

is because the T5D algorithm can reduce overestimation and enhance stability, allowing it to accurately estimate Q-values and optimize both discrete and continuous actions. It can appropriately adjust UAV trajectories, thus enabling it to modify strategies as the number of users changes, achieving optimal task completion times and SSEs. Additionally, as seen in Fig. 6(a), across different numbers of users, the schemes incorporating semantic communication technology consistently outperform those without it in both task completion times and SSEs, further demonstrating the advantages of semantic communication technology in the field of UAV-assisted MEC.

Fig. 7 depicts a comparative graph of task completion time and SSEs for various algorithms under different maximum jamming power levels. From the figure, it is observed that as the jamming power increases, the system's task completion time continuously grows, and the SSEs progressively decrease. This decrease is due to the jamming power reducing the SINR, which in turn lowers the semantic rate. The T5D algorithm consistently achieves the shortest task completion times and the highest SSEs across all jamming power levels, with minimal variation in its curve. This is because the T5D algorithm can reduce overestimation and maintains high stability, thereby allowing for optimal selection of UAV trajectorie, user associatios, and channel selection to collectively counter jamming attacks, minimizing their impact

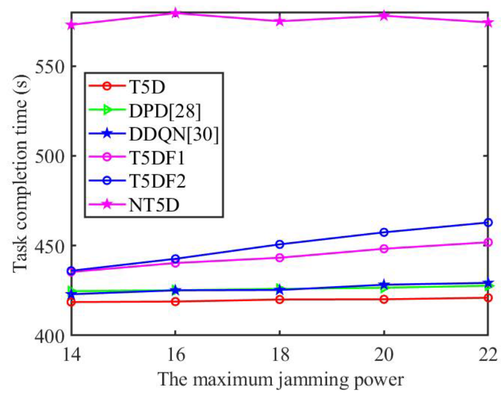


(a) Comparison of task completion time of different algorithms

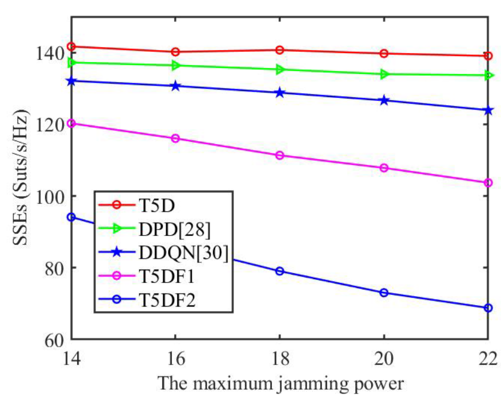


(b) Comparison of SSEs of different algorithms

Fig. 6. Performance comparison of different number of users.



(a) Comparison of task completion time of different algorithms



(b) Comparison of SSEs of different algorithms

Fig. 7. Performance comparison of different maximum jamming power.

on UAV-assisted MEC. The DPD and DDQN algorithms, due to poorer stability or limitations in optimizing discrete actions, fail to effectively coordinate various actions against the impact of jamming, resulting in greater curve fluctuations compared to the T5D algorithm. The T5DF1 and T5DF2 algorithms, with their restricted UAV trajectories, are significantly affected by jamming attacks, showing more noticeable fluctuations than the DPD and DDQN algorithms. From the discussion above, it is clear that the T5D algorithm effectively coordinates various actions to reduce the impact of jamming attacks, minimizing fluctuations in task completion time and SSEs curves. Compared to other algorithms, there is almost no visible change, indicating the excellent stability of the proposed algorithm under jamming attacks. Moreover, as seen in Fig. 7(a), under different jamming powers, implementations incorporating semantic communication technology consistently show significantly lower task completion times compared to those without it, demonstrating that semantic communication technology can effectively enhance UAV-assisted MEC performance under various jamming powers.

Fig. 8 compares the task completion time of different algorithms under various parameter multipliers. A parameter multiplier of 0.5 means that the parameters λ_1, λ_2, λ_3, and λ_4 are all multiplied by 0.5, and so forth. As can be seen, when the parameter multiplier is 1, the algorithm proposed in this paper achieves the optimal task completion time, indicating that the parameters selected in this paper are optimal. The main function of these selected parameters is to balance communication delay and computation delay while also enhancing semantic similarity, among other factors. When these parameters reach their optimal values, the best reward and minimal task completion time can be achieved. As the parameter multiplier changes, the task completion time of the T5D algorithm proposed in this paper remains the lowest, demonstrating that the proposed algorithm consistently achieves lower task completion times compared to other algorithms under different parameter multipliers, proving the effectiveness of this algorithm.

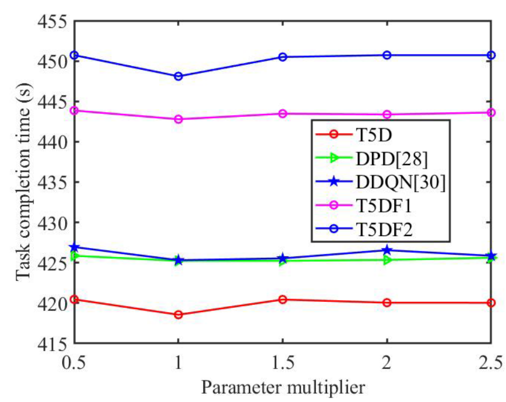


Fig. 8. Task completion time curves for different parameter multipliers.

From Fig. 4 to Fig. 8, it can be observed that the proposed T5D algorithm, integrating semantic communication, optimizes UAV trajectories, user associations, and channel selections strategies. This strategy outperforms the contrasted algorithms in terms of latency and SSEs. This indicates that the proposed T5D algorithm can effectively optimize UAV-assisted MEC strategies in a complex and dynamic jamming environment, particularly under minimum semantic similarity threshold.

VI. Conclusion

To optimize UAV-assisted MEC strategies in complex jamming environments, this paper proposes a DRL method named T5D, which is combined with semantic communication. The optimization objective is to minimize the task completion time and maximize SSEs under the minimum semantic similarity threshold. Given the complex, dynamic nature of communication networks in jamming environments, along with the vast state and action spaces, a DRL method is designed, leveraging the adaptability of DRL and the simplicity of semantic communication. This method demonstrates rapid convergence and effectively optimizes strategies such as UAV trajectories, user associations, and channel selections in the presence of jamming. Extensive simulation results indicate that the proposed T5D algorithm, combining semantic communication, outperforms benchmark algorithms in reducing task completion time, improving SSEs, and enhancing task offloading compared to non-semantic communication methods. In future work, we aim to explore the application of multi-agent DRL with semantic communication in the field of jamming-resistant UAV-assisted MEC. Additionally, we will explore how to reduce the additional requirements of semantic communication technology in data compression and processing, in order to enhance the encoding efficiency of semantic communication.

References

[1] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI models for green communications towards 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 210–247, 1st Quart., 2022.

[2] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2167–2191, 4th Quart., 2020.

[3] E. Coronado et al., "Zero touch management: A survey of network automation solutions for 5G and 6G networks," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2535–2578, 4th Quart., 2022.

[4] W. Yang et al., "Semantic communications for future internet: Fundamentals, applications, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 213–250, 1st Quart., 2022.

[5] Z. Lu et al., "Semantics-empowered communications: A tutorial-cum-survey," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 41–79, 1st Quart., 2024, doi: 10.1109/COMST.2023.3333342.

[6] Y. Wang et al., "Performance optimization for semantic communications: An attention-based reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2598–2613, Sep. 2022.

[7] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 9–39, Jan. 2023.

[8] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[9] Y. Yang et al., "Semantic communications with artificial intelligence tasks: Reducing bandwidth requirements and improving artificial intelligence task performance," *IEEE Ind. Electron. Mag.*, vol. 17, no. 3, pp. 4–13, May 2023.

[10] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, Aug. 2021.

[11] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, "Resource allocation for text semantic communications," *IEEE Wireless Commun. Lett.*, vol. 11, no. 7, pp. 1394–1398, Jul. 2022.

[12] J. Su et al., "Semantic communication-based dynamic resource allocation in D2D vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10784–10796, Mar. 2023.

[13] Z. Qin, F. Gao, B. Lin, X. Tao, G. Liu, and C. Pan, "A generalized semantic communication system: From sources to channels," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 18–26, Jun. 2023.

[14] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Ye Li, "QoE-aware resource allocation for semantic communication networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 3272–3277.

[15] J. Kang et al., "Personalized saliency in task-oriented semantic communications: Image transmission and performance analysis," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 186–201, Jan. 2023.

[16] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[17] W. Yang et al., "Semantic communication meets edge intelligence," *IEEE Wireless Commun.*, vol. 29, no. 5, pp. 28–35, Oct. 2022.

[18] N. C. Luong, Q.-V. Pham, T. Huynh-The, V.-D. Nguyen, D. W. K. Ng, and S. Chatzinotas, "Edge computing for semantic communication enabled metaverse: An incentive mechanism design," 2022, *arXiv:2212.06463*.

[19] W. Yu and J. Zhao, "Semantic communications, semantic edge computing, and semantic caching," 2023, *arXiv:2304.08096*.

[20] Z. Ji and Z. Qin, "Energy-efficient task offloading for semantic-aware networks," in *Proc. IEEE Int. Conf. Commun.*, Rome, Italy, Oct. 2023, pp. 3584–3589.

[21] P. Dong, Q. Wu, X. Zhang, and G. Ding, "Edge semantic cognitive intelligence for 6G networks: Novel theoretical models, enabling framework, and typical applications," *China Commun.*, vol. 19, no. 8, pp. 1–14, Aug. 2022.

[22] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[23] G. Geraci et al., "What will the future of UAV cellular communications be? A flight from 5G to 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1304–1335, 3rd Quart., 2022.

[24] H. Yang et al., "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nature Commun.*, vol. 13, no. 1, pp. 1723–2041, Jul. 2022.

[25] Q. Wu, M. Cui, G. Zhang, F. Wang, Q. Wu, and X. Chu, "Latency minimization for UAV-enabled URLLC-based mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3298–3311, Apr. 2024, doi: 10.1109/TWC.2023.3307154.

[26] D. Wang, Y. Jia, M. Dong, K. Ota, and L. Liang, "Blockchain-integrated UAV-assisted mobile edge computing: Trajectory planning and resource allocation," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1263–1275, Jan. 2024, doi: 10.1109/TVT.2023.3306740.

[27] H. Yang et al., "Intelligent reflecting surface assisted anti-jamming communications: A fast reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1963–1974, Mar. 2021.

[28] J. Chen et al., "Deep reinforcement learning based resource allocation in multi-UAV-aided MEC networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 296–309, Jan. 2023.

[29] S. Guo and X. Zhao, "Multi-agent deep reinforcement learning based transmission latency minimization for delay-sensitive cognitive satellite-UAV networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 131–144, Jan. 2023.

[30] X. Li, S. Cheng, H. Ding, M. Pan, and N. Zhao, "When UAVs meet cognitive radio: Offloading traffic under uncertain spectrum environment via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 824–838, Feb. 2023.

[31] 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.

[32] H. Hu, X. Zhu, F. Zhou, W. Wu, and R. Q. Hu, "Semantic-oriented resource allocation for multi-modal UAV semantic communication networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 7213–7218.

[33] L. Xiao, Y. Ding, J. Huang, S. Liu, Y. Tang, and H. Dai, "UAV anti-jamming video transmissions with QoE guarantee: A reinforcement learning-based approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5933–5947, Sep. 2021.

[34] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[35] L. Xiao, H. Li, S. Yu, Y. Zhang, L.-C. Wang, and S. Ma, "Reinforcement learning based network coding for drone-aided secure wireless communications," *IEEE Trans. Commun.*, vol. 70, no. 9, pp. 5975–5988, Sep. 2022.

[36] Y. Zhou et al., "Secure communications for UAV-enabled mobile edge computing systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 376–388, Jan. 2020.

[37] R. Karmakar, G. Kaddoum, and O. Akhrif, "A novel federated learning-based smart power and 3D trajectory control for fairness optimization in secure UAV-assisted MEC services," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4832–4848, May 2024, doi: 10.1109/TMC.2023.3298935.

[38] Y. Xu, T. Zhang, D. Yang, Y. Liu, and M. Tao, "Joint resource and trajectory optimization for security in UAV-assisted MEC systems," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 573–588, Jan. 2021.

[39] F. Song et al., "Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 12, pp. 7387–7405, Sep. 2022.

[40] S. Huang, L. Li, Q. Pan, W. Zheng, and Z. Lu, "Fine-grained task offloading for UAV via MEC-enabled networks," in *Proc. IEEE 30th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC Workshops)*, Istanbul, Turkey, Sep. 2019, pp. 1–6.

[41] L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji, and Y. Zhang, "Reinforcement learning-based mobile offloading for edge computing against jamming and interference," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6114–6126, Oct. 2020.

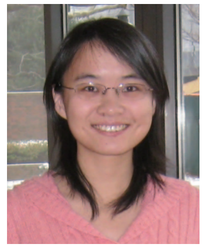
[42] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[43] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5872–5881.

Shuai Liu (Graduate Student Member, IEEE) received the M.S. degree in electricity theory and new technology from Shanghai University of Electric Power, Shanghai, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Information and Communication Engineering, Xiamen University. His research interests include UAV communications, anti-jamming, and reinforcement learning.

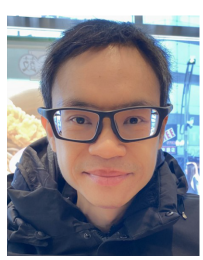
Helin Yang (Senior Member, IEEE) received the B.S. and M.S. degrees from the School of Telecommunications Information Engineering, Chongqing University of Posts and Telecommunications, in 2013 and 2016, respectively, and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2020. He is currently an Associate Professor with the School of Informatics, Xiamen University, Xiamen, China. His research interests include wireless communication and resource management.

Mengting Zheng (Student Member, IEEE) received the B.S. degree in communication engineering from Fuzhou University in 2023. She is currently pursuing the master's degree with the Department of Information and Communication Engineering, Xiamen University. Her research interests include edge computing, UAV communications, and resource scheduling.



Liang Xiao (Senior Member, IEEE) received the B.S. degree in communication engineering from Nanjing University of Posts and Telecommunications, China, in 2000, the M.S. degree in electrical engineering from Tsinghua University, China, in 2003, and the Ph.D. degree in electrical engineering from Rutgers University, NJ, USA, in 2009. She was a Visiting Professor with Princeton University; Virginia Tech; and the University of Maryland, College Park. She is currently a Professor with the Department of Information and Communication Engineering, Xiamen University, Xiamen, China. She was a recipient of the Best Paper Award for 2016 INFOCOM Big Security WS and 2017 ICC. She has served as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and a Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING.

Zehui Xiong (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore. He is currently an Assistant Professor with Singapore University of Technology and Design; and also an Honorary Adjunct Senior Research Scientist with Alibaba-NTU Singapore Joint Research Institute, Singapore. He was a Visiting Scholar with Princeton University and the University of Waterloo. He has published more than 200 research papers in leading journals and flagship conferences and many of them are ESI highly cited papers. His research interests include wireless communications, the Internet of Things, blockchain, edge intelligence, and metaverse. He has won over ten best paper awards in international conferences and is listed in the World's Top 2% Scientists identified by Stanford University. He was a recipient of the IEEE Early Career Researcher Award for Excellence in Scalable Computing, the IEEE Technical Committee on Blockchain and Distributed Ledger Technologies Early Career Award, the IEEE Internet Technical Committee Early Achievement Award, the IEEE TCSVC Rising Star Award, the IEEE TCI Rising Star Award, the IEEE TCCLD Rising Star Award, the IEEE Best Land Transportation Paper Award, the IEEE CSIM Technical Committee Best Journal Paper Award, the IEEE SPCC Technical Committee Best Paper Award, the IEEE VTS Singapore Best Paper Award, the Chinese Government Award for Outstanding Students Abroad, and the NTU SCSE Best Ph.D. Thesis Runner-Up Award. He is serving as the Associate Director of Future Communications Research and Development Program. In 2023, he was featured on the list of Forbes Asia 30 under 30. He is serving as an Editor or a Guest Editor for many leading journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.



Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada. He is a Professor with the College of Computing and Data Science, Nanyang Technological University, Singapore. His research interests are in the areas of mobile generative AI, edge intelligence, decentralized machine learning, and incentive mechanism design.