Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Autonomous decision-making of UAV cluster with communication constraints based on reinforcement learning

Zhang Ting-Ting[1,5*], Chen Yan[1], Dong Ren-zhi[2], Chen Tao[3], Liu Yan[4], Zhang Kai-Ge[4], Song Ai-Guo[5] and Lan Yu-Shi[6]

Abstract

Artificial intelligence techniques are increasingly applied in the study of autonomous decision-making in unmanned clustered distributed systems. However, communication constraints has become a big bottleneck that restricts its performance. To address the need for unmanned aerial vehicles(UAVs) to execute collaborative attack missions in complex communication-constrained environments, this paper propose an autonomous decision-making method for UAVs based on Multi-Agent Reinforcement Learning (MARL). Firstly, the autonomous decision-making processes of UAV clusters are modeled as Decentralized Partially Observable Markov Decision Processes(Dec-POMDPs). Next, the algorithm is enhanced within the framework of Multi-Agent Deep Deterministic Policy Gradient(MADDPG) by designing an explicit inter-intelligent communication mechanism to achieve information exchange among UAVs. Subsequently, the algorithm utilizes Long Short-Term Memory(LSTM) networks to process the local observations of the UAVs, enhancing the effectiveness of the information sent by combining historical data with current observations. Finally, multiple rounds of experiments are conducted across various communication-constrained scenarios. Simulation results indicate that the proposed method improves the task completion capability by 46.0% and enhances stability by 24.9% compared to baseline algorithm MADDPG. Additionally, the algorithm demonstrates better generalization and exhibits good scalability, effectively adapting to varying numbers of UAVs. This research provides new theoretical insights and a technical framework for the collaboration of UAVs in environments with communication constraints, which holds great practical importance in improving the ability and application scope of UAV systems.

Keywords UAV cluster, Autonomous decision-making, Communication constraint, Reinforcement learning

Introduction

In the rapidly advancing field of artificial intelligence technology [1–6], the applications of UAVs have expanded from military reconnaissance to include civilian monitoring and disaster relief [7–9]. These UAVs possess unique advantages in executing complex tasks, particularly in environments that pose high risks or are otherwise inaccessible to humans. Their operational efficiency and safety have been significantly enhanced due to their flexibility and operability. However, the complex and dynamic nature of the real world presents significant challenges to the application of UAV technology, with

*Correspondence:
Zhang Ting-Ting
101101964@seu.edu.cn
[1] Army Engineering University of PLA, College of Command and Control Engineering, Nanjing, Jiangsu, China
[2] Cetccloud (Beijing) Technology Co, Nanjing, Jiangsu, China
[3] National University of Defense Technology, Changsha, Hunan, China
[4] North Automatic Control Technology Institute, Taiyuan, Shanxi, China
[5] Southeast University, Nanjing, Jiangsu, China
[6] Nanjing Research Institute of Electronic Engineering, Nanjing, Jiangsu, China

communication constraints being a particularly pressing issue [10–12].

Communication constraints encompass a range of scenarios that can affect the performance and reliability of UAVs. As shown in Fig. 1, this study considers two specific forms of communication constraints: the limitation of the UAV communication radius and the presence of external interference obstacles. The limitation of communication radius refers to the inherent technical constraint that the propagation of wireless signals restricts the communication capability of a UAV to a specific physical distance. In wide geographic areas, such as oceans or remote mountainous regions, a UAV may exceed this communication radius and lose contact with the control center or other UAVs, which may result in mission interruption or failure. External interference obstacles include various external factors, such as electromagnetic interference, signal blocking, or interference from other radio equipment, that may disrupt UAV communication signals in certain environments, such as urban canyons or battlefield settings. Such interference may lead to communication interruptions or data transmission errors [13].

Currently, several studies are addressing the challenges of UAV path planning while considering communication constraints. Chen et al. [14] proposed a multi-UAV path planning method based on Deep Q Networks to tackle the collaborative coverage problem while adhering to UAV communication distance constraints. Xiao et al. [15] improved upon traditional target probability graph updating methods to achieve multi-UAV cooperative target search under limited communication distances. Chen et al. [16] designed a formation control method based on virtual long-range state estimation, although it still assumes a fixed communication topology for the UAV cluster. Cao et al. [17] addressed the planning problem for multi-base multi-UAV cooperative search for multiple targets under communication limitations. Yu et al. [18] assumed that the communication success rate between UAVs is a probability related to distance, and under this condition, proposes a method based on D3QN to solve the multi-target rescue problem. Fu et al. [19] designed a collision prediction mechanism and artificial potential function to address obstacle avoidance in cluster formation control under limited communication radii. Bramblett et al. [20] considered the existence of communication interference zones and proposes a new epistemic planning approach to enable multi-robot search and rescue under communication constraints.

This research recognizes that the two scenarios illustrated in Fig. 1 are typical examples of communication constraints that UAVs may encounter. The focus of this paper is on addressing the challenges associated with these scenarios. To address these challenges, reinforcement learning, specifically MARL, is employed as a potential solution in this study [21, 22]. Reinforcement learning allows agents to learn optimal policies through interaction with the environment, without requiring precise modeling, which makes it particularly suitable for dynamic and uncertain conditions [23]. MARL enhances this approach by emphasizing collaborative learning of multiple agents in a shared environment [24–26], which is particularly relevant for the coordinated control of UAV clusters under communication constraints. It facilitates global collaboration by enabling UAVs to make independent decisions while considering the actions of other UAVs.

This paper proposes an intelligent decision-making method for UAVs based on the MARL framework, aiming to improve the mission execution capability of UAV clusters in constrained environments through local information exchange and cooperation.

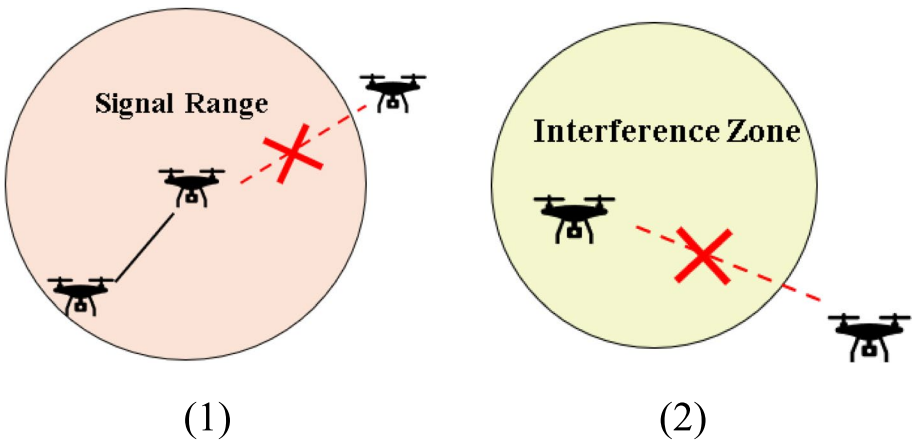

Fig. 1 Communication constraints

Problem definition

In this study, each UAV is equipped with on-board sensors that have a limited sensing range. The action-state transition of the UAVs follows a Markovian process, which allow us to model the decision-making process of the UAV cluster using Dec-POMDPs [27, 28]. In the Dec-POMDPs framework, each UAV is treated as an independent intelligent agent. These agents make decisions based on their acquired information such as local observations, while the joint action of all agents determines the final global return. Therefore, finding the optimal policy in the joint action space is the core challenge of the problem. The decision-making problem of a UAV cluster based on Dec-POMDPs can be represented by a tuple $< I, S, A, O, R, p, \gamma >$, where the elements are denoted:

(1) Index

$I = 1, \cdots, n$ is the collection of UAVs on a mission and n is the number of UAVs.

(2) State

S is the set of possible global states where agents exist at each time step. Each UAV's state includes its own position information and its velocity.

(3) Action

$A = \times_{i=1}^{n} A_i$ is the set of joint action. The joint action at any given moment consists of the action of all UAVs in the cluster, and the joint action $\boldsymbol{a} =< a_1, \cdots, a_n > \in A$,$a_i \in A_i$, A_i is the set of actions of UAV_i.

(4) Observation space

$O = \times_{i=1}^{n} O_i$ is the observation space for the agents. It is common to utilize the observation O as an approximation for the underlying true state s in practice.

(5) Reward function

$R = \sum_{i=1}^{n} R_i : S \times A \rightarrow R$ describes the process by which a cluster of UAVs takes joint action $\boldsymbol{a}$ at global state s to obtain a reward.

(6) Transition probability

$p(s(t+1)|s(t), \boldsymbol{a}(t)) : S \times A \times S \rightarrow R^{+}$ represents the probability distribution of state transition given the current global state s and joint action $\boldsymbol{a}$.

(7) Discount factor

$\gamma \in [0,1]$ is used to calculate cumulative rewards. Under the Dec-MOMDPs model, each UAV is trained to learn a policy $\mu_i(a_i|o_i) : O_i \rightarrow A_i$ that maximises the expected value of the cumulative reward $J(\mu_i) = E\left[\sum_{t=0}^{\infty} \gamma^t R_i\right]$. R_i is the reward obtained by the UAV at the current moment.

By exchanging information with other UAVs in the cluster, UAV_i can overcome the limitations of its own sensory capabilities, thereby more effectively supporting its decision-making process. The communication-based autonomous decision-making model of UAV_i can be represented as $\mu_i(a_i|o_i, m_{-i}) : O_i \times M_{-i} \rightarrow A_i$,where $m_{-i} =< m_1, \cdots, m_{i-1}, m_{i+1}, \cdots, m_n > \in M_{-i}$ denotes the messages received by UAV_i from other UAVs in the cluster.

To describe the maneuvering behavior and state transition process of an UAV, we assume that the flight altitude of the UAV is constant. As shown in Fig. 2, the state of UAV_i is denoted as $s_i(t) = [p_i(t), v_i(t)]^T$, $p_i = [x_i, y_i]$ represents the position of UAV_i in a two-dimensional inertial coordinate system, and $v_i(t)$ represents the velocity. The approximate discrete dynamics model of UAV_i from time t to time $t+1$ is as follows:

$$x_i(t+1) = x_i(t) + v_i(t)\Delta t \cos\phi_i(t) \tag{1}$$

$$y_i(t+1) = y_i(t) + v_i(t)\Delta t \sin\phi_i(t) \tag{2}$$

$$v_i(t+1) = v_i(t) + u_i(t)\Delta t \tag{3}$$

$$\phi_i(t+1) = \phi_i(t) + \omega_i(t)\Delta t \tag{4}$$

Whereϕ_i, $u_i(t)$, $\omega_i(t)$ are the coarse angel, forward acceleration, and turning acceleration of UAV_i at time t respectively, Δt is the time step, and the action vector of UAV_i is denoted as $a_i = [u_i(t), \omega_i(t)]^T$.

Training method for UAVs' policy models

During a mission, UAV_i needs to decide what information to send at each moment and chooses the actions to execute based on the received information and its own state. The selection of these actions relies on its policy model. If the policy model $\mu_i(a_i|o_i, m_{-i})$ is known, the cumulative reward associated with this policy can be estimated, with larger reward values indicating a better policy model.

Choosing an appropriate training method for UAVs' policy models is crucial. Deep Reinforcement Learning (DRL) techniques [29], particularly MADDPG, establish a MARL framework that leverages reward information to provide continuous feedback on agent actions during training

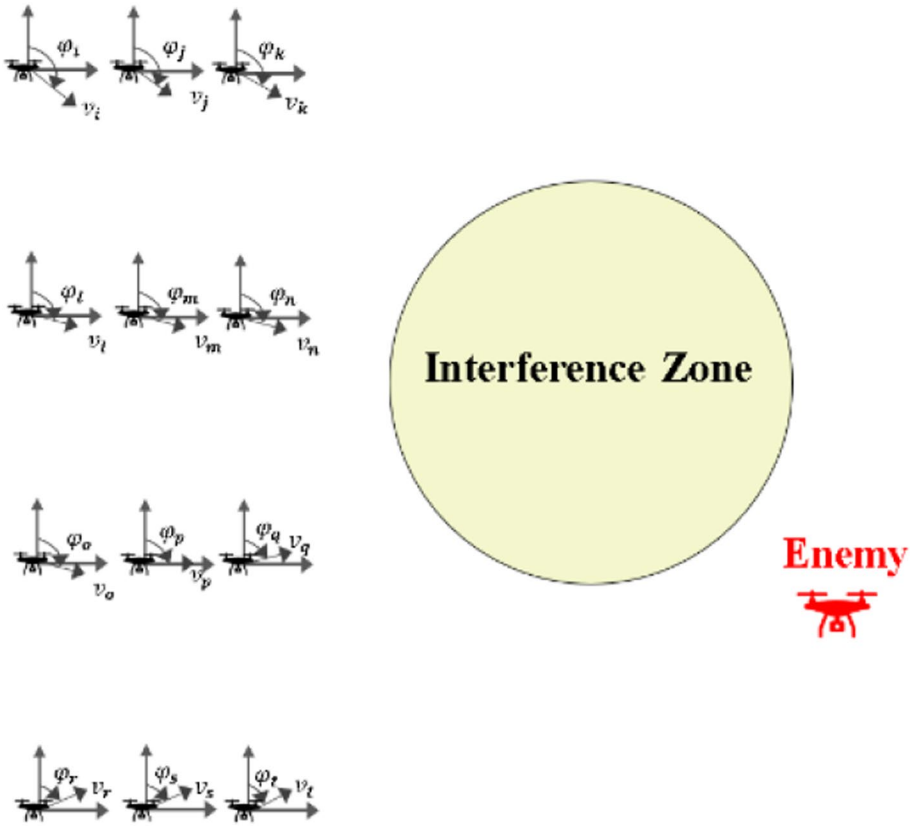

Fig. 2 UAVs' kinematic behavior

[30]. This enables the acquisition of knowledge regarding decision-making for behavior policies. This approach can achieve optimal policy solutions even in the presence of dynamic external environmental changes and uncertain noise, without the need for pre-established data on system states or knowledge of behavioral relationships [31–33]. However, the original MADDPG algorithm does not incorporate explicit communication between agents; it relies solely on a centralized Critic network to obtain global information, thus enabling implicit communication. Inspired by the aforementioned methods, we designed our method within the MADDPG framework, as illustrated in Fig. 3. UAV_i acquires its observations o_i from the environment and encodes these observations along with its historical state information using LSTM and Multi-layer Perceptron (MLP) layers. This process produces the message m_i that it decides to send at the current moment, while also receiving messages m_{-i} from other UAVs. LSTM excels in modelling sequential data and is particularly suited to capturing long-term dependencies in time series. The use of LSTM networks allows for better handling of local observations from UAVs and improves the effectiveness of the messages sent by combining historical information with current observations. By integrating its local information with the received messages, UAV_i determines its optimal action at the current time step.

Model training methods based on limited communications

As shown in Fig. 4, the policy models of the UAV cluster are trained using the Actor-Critic framework. The Actor network simulates the policy function μ with parameters θ_{μ}. The Critic network is a neural network simulation of the state evaluation function $Q(s,a)$ with parameters θ_{Q}.

At a given moment, UAV_i takes as input its own perceptual information o_i from global state s, forms its own messagem_i and receives messages m_{-i} from other UAVs. UAV_i select action based on its policy to form the joint action a and receives feedback rewards R_i. The state transition process is stored as state transfer data $< s, a, s', R >$ in the experience buffer D for Critic network training.

During training, the Critic network of UAV_i randomly draws a batch of state transfer data from the experience buffer to update the current Critic network.

$$L(\theta_{Q_i}) = \frac{1}{T}\left[\sum_j \left(y^j - Q_i^{\mu_i}\left(s^j, a^j\right)\right)^2\right] \tag{5}$$

where $y^j = R_i^j + \gamma\, Q_i^{\mu_i}\left(s'^j, a_1, \cdots, a_n\right)\Big|_{a_i=\mu_i(o_i,m_{-i})}$.

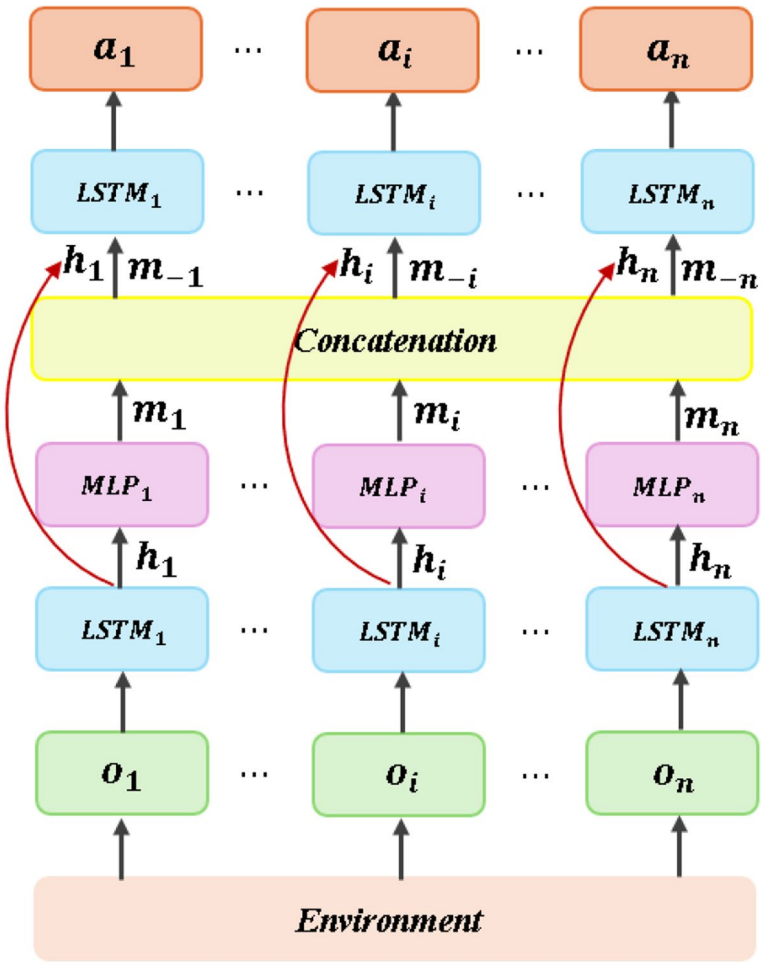

Fig. 3 Overview of our method

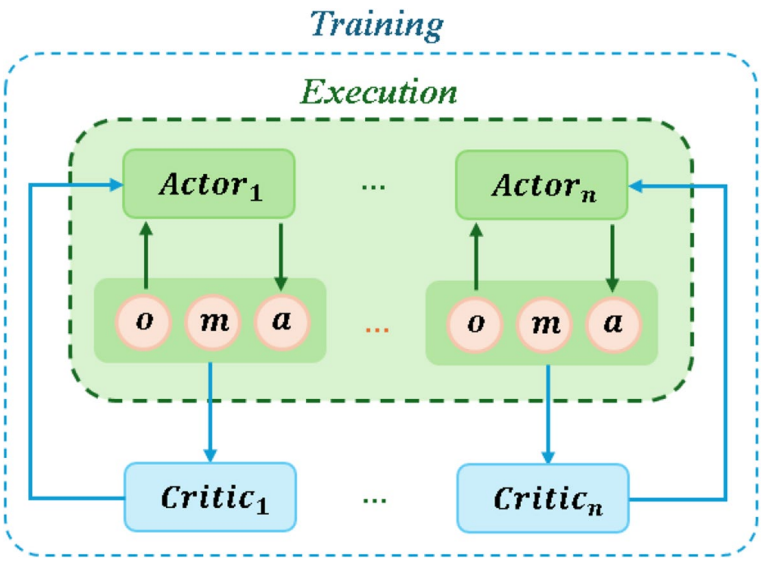

Fig. 4 Training and executing process

The Actor network is updated according to the estimate of Critic network.

$$\nabla_{\theta_{\mu_i}} J(\theta_{\mu_i}) = \mathrm{E}\left[\nabla_{\theta_{\mu_i}} \ln \pi_{\theta_{\mu_i}}(a^i|s) Q^{\mu_i}(s,a)\right] \quad (6)$$

To avoid overestimating action values and to enhance training stability, target networks are introduced during the training process. The parameters of the target networks are updated gradually based on the current networks. Algorithm 1 outlines the procedure of the algorithm proposed in this paper.

Algorithm 1 Our proposed algorithm

Initialize each UAV's Actor network parameter θ_{μ_i}, Critic network parameter θ_{Q_i}, and the corresponding target Actor network parameter $\theta_{\mu_i}^{\text{target}}$ and target Critic network parameter $\theta_{Q_i}^{target}$, initialize soft update parameter $\tau \in (0,1)$.	
1	For episode =1 to MaxEpisode do
2	Random initialization of the initial state of the UAV cluster
3	For each UAV_i get observation o_i and send message m_i
4	For each UAV_i receive message m_{-i} and select action $a_i = \mu_i(o_i, m_{-i})$
5	UAV cluster perform joint action $a = < a_1, \cdots, a_n >$
6	Get reward R and transfer to the state s'
7	Restore the state transition data $< s, \boldsymbol{a}, s', R >$ into the experience buffer D
8	For UAVs $i = 1$ to n do
9	Randomly selected T samples from the experience buffer
10	Calculating the loss function
11	Update θ_{Q_i} based on loss function
12	Update θ_{μ_i}
13	End For
14	For UAVs $i = 1$ to n do
15	$\theta_{Q_i}^{target} \leftarrow \tau\theta_{Q_i} + (1-\tau)\theta_{Q_i}^{target}$
16	$\theta_{\mu_i}^{\text{target}} \leftarrow \tau\theta_{\mu_i} + (1-\tau)\theta_{\mu_i}^{\text{target}}$
17	End For
18	End For

Once trained, the policy model μ_i of UAV_i can choose its action based on its own observation information o_i and messages m_{-i}received from other UAVs.

Simulation experiments

To assess the effectiveness of the proposed algorithm, this study creates customized experimental environments using the open-source multi-agent environment provided by OpenAI to simulate UAV engagements in communication-constrained combat scenarios [30].

Experimental introduction

In a communication-constrained simulation environment, our UAVs must combine their own observations with limited communication to conduct collaborative attacks on a single enemy UAV under various circumstances. The effectiveness of the mission completion is measured by counting the number of collaborative attacks performed by our UAVs on the enemy UAV within a limited time frame. We consider more than 150 collaborative attacks within a constrained number of time steps to be indicative of excellent performance. The experimental setup is shown in Table 1.

(1) State space

The state space is categorized into local and global states. The local state refers to the data within the sensing range of the on-board sensors, while the global state represents the overall state of the environment. However, due to partial observability, each UAV can only access local states and cannot fully perceive the global state.

(2) Action space

The action space of each UAV includes parameters such as forward acceleration and turning acceleration.

(3) Reward function

The objective of the reward function is to enable UAVs to learn how to effectively accomplish combat missions under communication constraints.

The objective of our UAVs is to collaboratively attack the enemy and maximize rewards. When two or more of our UAVs encounter an enemy UAV, the attacking UAVs receive positive rewards. Additionally, to guide our UAVs toward enemy UAVs for an attack, we provide rewards based on the real-time distance between our UAVs and the enemy UAVs. Collaborative attacks necessitate that our UAVs maintain a relatively close distance to the opponent. Without distance-based rewards, it becomes challenging for the UAVs to explore subsequent rewards for collaborative attacks. Therefore, we designed the following reward function as shown in Table 2.

Where N_{attack} represents the number of our UAVs participating in a coordinated attack on the enemy at this timestep, while p_i and p_{enermy} denote the positions of our UAV_i and the enemy UAV, respectively, at this timestep.

Experimental setup

(1) Experimental hardware environment.

Table 1 Experimental setup

Name	Value
Map size	20km*20km
Time limits	10min
Maximum speed	60m/s
Signal range	5km
Interference zone radius	4km

Table 2 Reward function

	Description	Calculation
R_{coor}	Reward for coordinated attacks	$R_{coor} = \frac{N_{attack} * \left(N_{attack} - 1\right)}{2}$
$R_{dis,i}$	Reward for moving towards enemy	$R_{dis,i} = -\sqrt{\left(p_i - p_{enermy}\right)^2}$

Table 3 Experiment-related parameter settings

Parameters	Name	Value
γ	Discount factor	0.95
Batch-Size	Number of batch samples	128
Num-Episode	Number of training episodes	4000
Max-Episode-Len	Maximum episode length	25
α_μ	Actor network learning rate	0.005
α_μ	Critic network learning rate	0.01

The training process was conducted on a server equipped with 32 GB of RAM and an Intel i5-10400 processor, featuring a GeForce RTX 4060 Ti graphics card.

(2) Experimental software environment

The training process utilized the Windows 10 operating system and the Python 3.6 interpreter. The environment dependencies of the algorithm include OpenAI Gym 0.10.5, TensorFlow 1.8.0, and NumPy 1.14.5. Each LSTM layer and MLP layer consists of 128 hidden units.

(3) Training parameters

The training parameters are set as detailed in Table 3.

(4) Experimental methods

To validate the effectiveness of the proposed algorithm in achieving the collaborative capabilities of UAVs in communication-restricted scenarios, we designed a series of experimental scenarios with communication constraints for multiple rounds of testing, using the MADDPG algorithm as the baseline. Subsequently, we analyzed the experimental results from various perspectives.

Experiment 1: scenarios with limitation of communication radius

Case 1: Our four UAVs and enemy UAV have fixed initial positions.

Case 2: The initial positions of our UAVs and the enemy UAV are randomly generated in the upper-left and lower-right areas of the map, respectively.

Case 3: The initial positions of our UAVs and the enemy UAV are randomly generated in the left and right areas of the map, respectively.

Case 4: The initial positions of our UAVs and the enemy UAV are randomly generated within the map.

Experiment 2: scenarios with external interference obstacles

In this experiment, our four UAVs and the enemy UAV have fixed initial positions, while the positions of the external interference obstacles vary across four different cases. From Case 1 to Case 4, the difficulty of the experimental scenarios gradually increases, and the position of the interference zone shifts from the edge of the map in Case 1 to the middle between our UAVs and the enemy UAV in Case 4.

Experiment 3: scenarios with mixed communication constraints

In this experiment, there are two types of communication constraints: the limitation of a UAV's communication radius and interference zones. In each case, the initial positions of our four UAVs and the enemy UAV are fixed. The position of the interference zone is the same as in Experiment 2, gradually shifting from the edge of the map in Case 1 to the middle between our UAVs and the enemy UAV in Case 4.

Simulation results and analysis

We analyze the effectiveness of our proposed method in maintaining the collaborative capabilities of UAVs in communication-restricted scenarios from multiple perspectives, including the completion of collaborative attack tasks, reward values, generalization ability, scalability, algorithm stability, and the convergence speed of training.

Completion of collaborative attack tasks

The cumulative number of collaborative attacks performed by the algorithms serves as an intuitive indicator of their task completion status. As shown in Fig. 5 and Fig. 6, in Experiment 1, Experiment 2 and Experiment 3, the cumulative number of collaborative attacks achieved by our proposed algorithms exceeds 150, indicating that the algorithms are able to effectively complete the collaborative attack tasks in different scenarios, and show a strong task execution capability.

In Experiment 1, Case 1, the average cumulative numbers of collaborative attacks achieved by our algorithm and the baseline algorithm MADDPG were 203.6 and 150.2, respectively, showing effective task completion. In

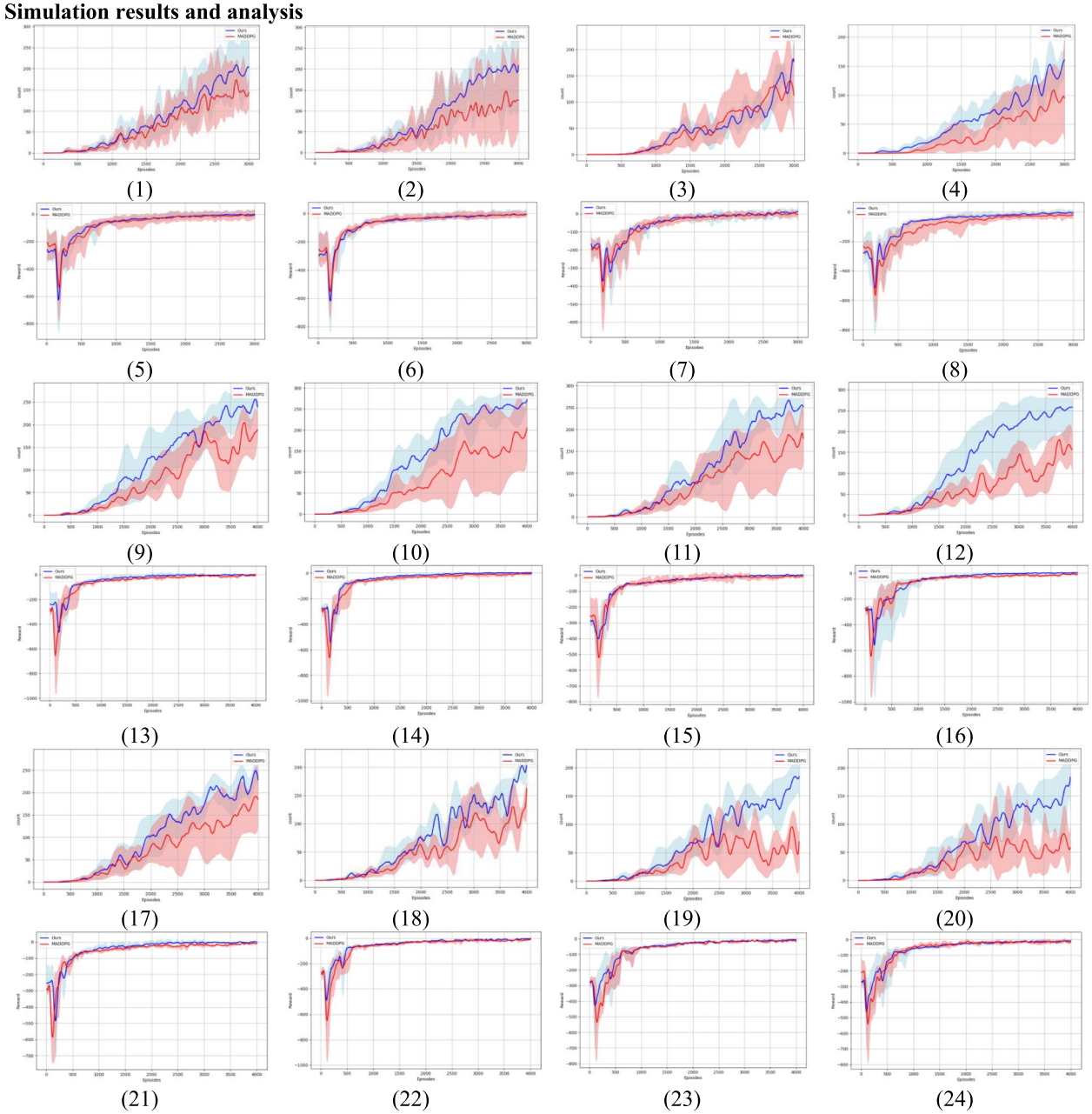

Fig. 5 Learning curves of the UAVs in different scenarios. (1)-(4) represent the learning curves of the cumulative number of collaborative attacks for the four cases of Experiment 1, while (5)-(8) represent the learning curves of rewards for the four cases of Experiment 1. (9)-(12) represent the learning curves of the cumulative number of collaborative attacks for the four cases of Experiment 2, (13)-(16) represent the learning curves of rewards for the four cases of Experiment 2. (17)-(20) represent the learning curves of the cumulative number of collaborative attacks for the four cases of Experiment 3, (21)-(24) represent the learning curves of rewards for the four cases of Experiment 3

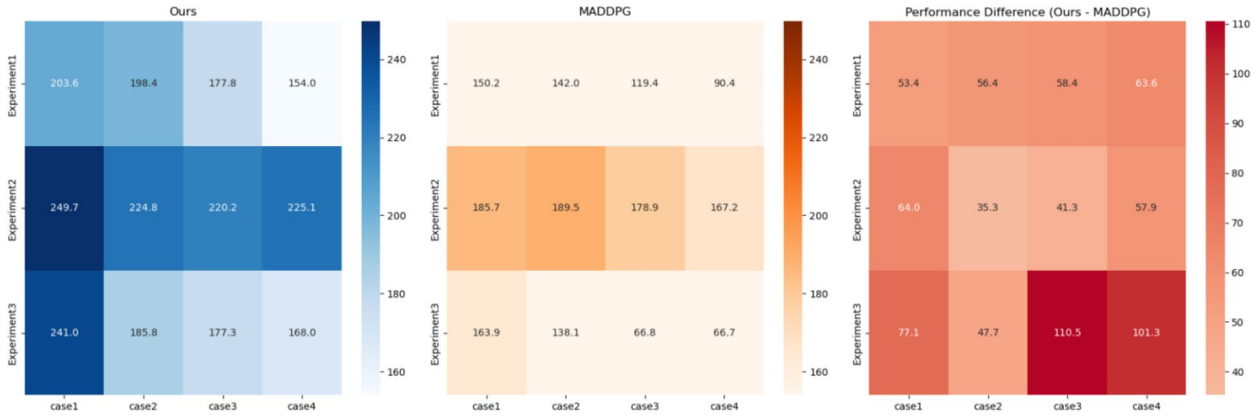

Fig. 6 The heatmap representation of the average cumulative number of collaborative attacks for our algorithm and the baseline algorithm MADDPG in each case of the three experiments

the more complex cases 2, 3, and 4, our algorithms also achieve an average cumulative number of collaborative attacks of over 150. In contrast, while MADDPG is able to complete the task in the simpler scenarios of Case 1, its performance drops significantly as the difficulty of the scenarios increases, and the average cumulative number of attacks in Cases 2, 3, and 4 falls below the standard of excellence of 150 collaborative attacks.

In Experiment 2, both our algorithm and the baseline algorithm, MADDPG, achieve an average cumulative number of collaborative attacks that exceeds the criterion of 150 in all four scenarios, indicating that both algorithms are capable of completing the collaborative attack task under the settings of Experiment 2. A numerical comparison between the two shows that our algorithm consistently outperforms the baseline algorithm.

In the hybrid scenario of Experiment 3, the baseline algorithm MADDPG is only able to complete the collaborative attack task in the simplest case, Case 1. In the more challenging Cases 2, 3, and 4, the average cumulative number of collaborative attacks achieved by MADDPG drops significantly below the threshold of 150, with performance decreasing as the difficulty increases. In contrast, our algorithm shows good performance in all four cases, with the average cumulative number exceeding the threshold of 150 coordinated attacks. Furthermore, as shown in Fig. 5, the performance of MADDPG significantly declines in the most complex Cases 3 and 4, with a noticeable gap compared to our proposed algorithm.

By aggregating the statistics from Experiments 1, 2, and 3, we find that our algorithm achieves an average cumulative number of collaborative attacks 46.0% higher than the benchmark algorithm, MADDPG, which indicates a substantial improvement in our task completion capabilities.

Reward values

As shown in Fig. 5 (5)-(8), (13)-(16), and (21)-(24), in the scenarios of each case from Experiments 1 to 3, our proposed algorithm effectively learns policies throughout the training process, resulting in continuously increasing reward values, which are higher than those of the baseline algorithm, MADDPG.

Generalization capabilities

As described in the experimental setup, in Experiment 1 Case 1, the initial positions of our four UAVs and the enemy UAV are fixed. In contrast, in Cases 2, 3 and 4, the initial positions of our UAVs and the enemy UAV are randomly generated with increasing randomness leading to higher uncertainty. Consequently, we evaluate the generalization ability of our proposed algorithm by analyzing the results of Experiment 1.

As shown in Fig. 7 and Fig. 8, the average cumulative number of collaborative attacks achieved by our proposed algorithm exceeds the excellence threshold of 150 in the random scenarios of Cases 2, 3, and 4. This indicates that our proposed algorithm is adaptable and can effectively perform the collaborative attack task in these random scenarios.

Furthermore, by analyzing and comparing the results of Case 1 (initial position fixed) and Case 4 (position completely random) (203.5 and 154.0, respectively), we find that the average cumulative number of collaborative attacks achieved by our algorithm in the completely random case is at least 75% of that in the case where the initial position is fixed. All these results show that our proposed algorithm has a strong generalization ability.

Scalability

To test the scalability of our algorithm, we conducted Experiment 4 using scenarios that included a greater

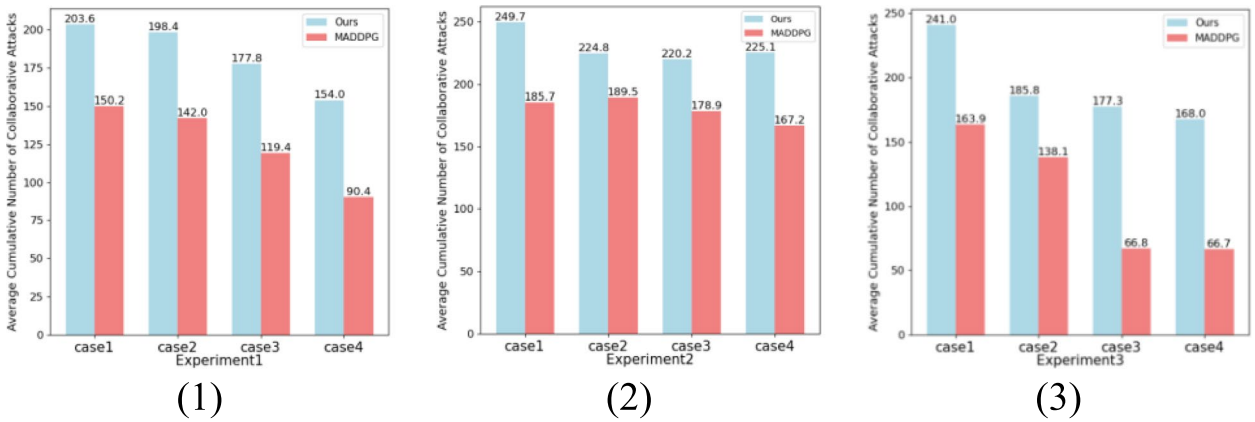

Fig. 7 Average cumulative number of collaborative attacks in Experiment 1, 2, and 3

number of UAVs. In this experiment, we added more UAVs (8 and 12) while still incorporating both types of communication restrictions. The initial positions of our UAVs and the enemy UAVs are randomly generated in the left and right areas of the map, respectively.

As the number of UAVs increases, the complexity and uncertainty of the tasks also rise, which typically presents challenges for algorithm performance. As shown in Fig. 9, when the number of our UAVs was increased to 8 and 12, our algorithm still achieved good results after training. Although more training episodes were required for convergence when the number of UAVs reached 12, the overall results indicate that our algorithm possesses a certain degree of scalability. It can adapt to these variations and effectively operate in larger-scale and more complex scenarios. This characteristic not only ensures the practical applicability of the algorithm but also lays the groundwork for future applications in more complex environments.

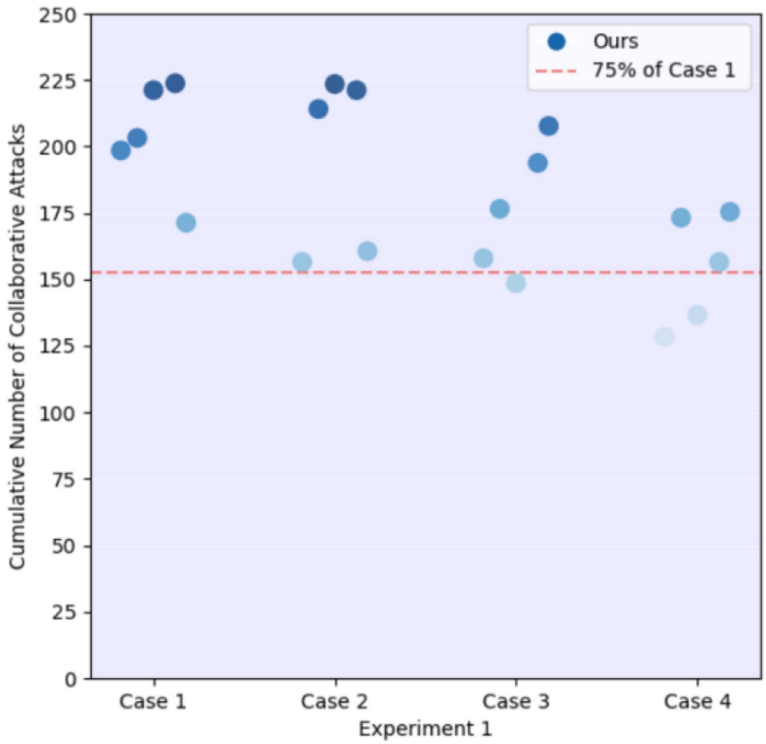

Fig. 8 Cumulative number of collaborative attacks across multiple rounds in Experiment 1

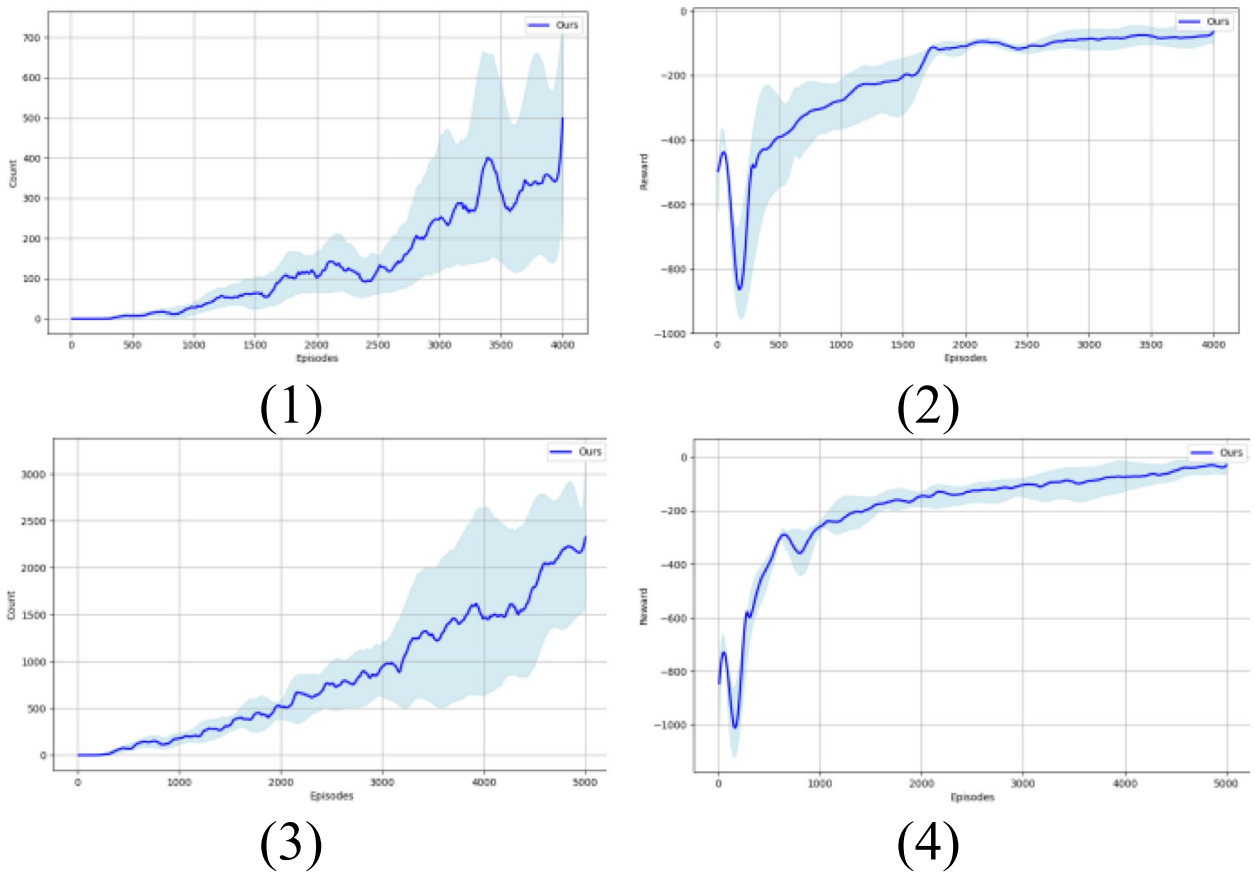

Fig. 9 (1)-(2) and (3)-(4) represent the learning curves of the cumulative number of collaborative attacks and reward values when the number of our UAVs is 8 and 12, respectively, in Experiment 4

Algorithm stability

Stability is a key factor in evaluating algorithms; therefore, we assessed the stability of the algorithms by analyzing the fluctuations observed in multiple rounds of experiments. As shown in Fig. 10, we collected data on the cumulative number of the last 100 coordinated attacks in each round of experiments and calculated the range (the difference between the maximum and minimum values) of each attack over these rounds. The results indicate that the fluctuations of our algorithm are consistently lower than MADDPG for all scenarios in Experiments 1, 2, and 3. Although the fluctuations increase with the complexity of the scenarios, the maximum value remains lower than that of MADDPG. By aggregating the fluctuation values for all scenarios in Experiments 1, 2, and 3, we find that the average fluctuation of our algorithm is about 24.9% lower than that of MADDPG, which suggests that the stability is improved by about 24.9%.

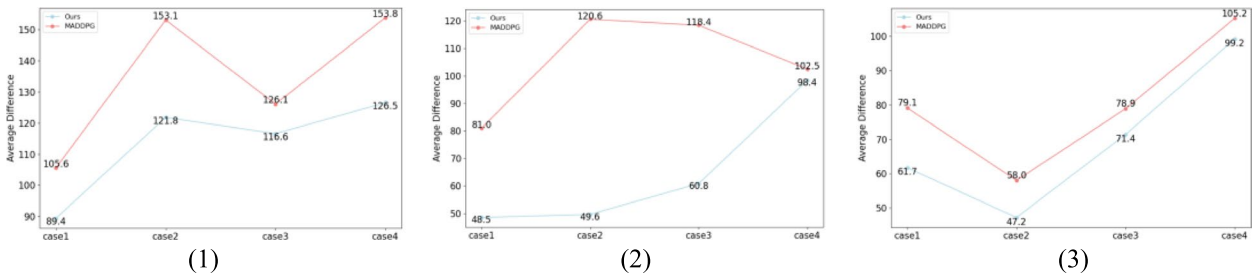

Fig. 10 Average difference between the upper and lower bounds of cumulative number of collaborative attacks in Experiment 1,2, and 3

Convergence speed

Although our algorithm has been modified within the framework of MADDPG, increasing its complexity, its convergence speed is not inferior to that of MADDPG. By comparing the training curves from multiple rounds of experiments across Cases 1, 2, and 3, as shown in Fig. 5 (1) to (24), this conclusion is validated.

Specifically, despite the increased structural and computational complexity of our algorithm, its training curves exhibit a convergence trend similar to that of MADDPG. This indicates that, while complexity has been added, our approach remains effective in optimizing policies during training, allowing it to quickly approach optimal solutions. This performance not only reflects the efficacy of the algorithm but also demonstrates its adaptability in complex task environments. Such capabilities instill confidence in its application in more complex scenarios in the future, suggesting that it can achieve a favorable balance between performance and efficiency.

As shown in Fig. 11, the UAVs on both sides were initially randomly generated on the map. After extensive training, our drones learned to initially navigate towards the centre of the map while avoiding interference areas as much as possible in order to mitigate the limitations imposed by excessive communication distances. At some point after the UAV rendezvous, one of our UAVs detected the location of an enemy UAV. It encoded its observation and sent it to other friendly UAVs within communication range. Subsequently, our UAVs adjusted their flight direction and speed according to the position of the enemy and friendly UAVs, steadily approached the enemy, and conduct coordinated attacks until the end of the episode.

In conclusion, by analyzing the above aspects, we conclude that our proposed algorithm can effectively accomplish the collaborative task, demonstrating strong generalization ability, scalability and stability, without compromising the convergence speed.

Conclusion

Aiming at the needs of UAVs to perform collaborative attack tasks in complex and communication-constrained environments, this study proposes an intelligent decision-making method based on MARL. The method facilitates effective UAV collaboration in restricted environments by exchanging local information via communication to ensure smooth execution of collaborative attack tasks. Simulation experiments show that the method outperforms the traditional baseline algorithm in several key metrics, with a 46.0% improvement in task completion rate and a 24.9% improvement in stability. Furthermore, the method exhibits high scalability and effectively accommodates varying numbers of UAVs. This study provides new theoretical insights and technical frameworks for cooperative UAV operations in communication-constrained scenarios, which is of great practical significance

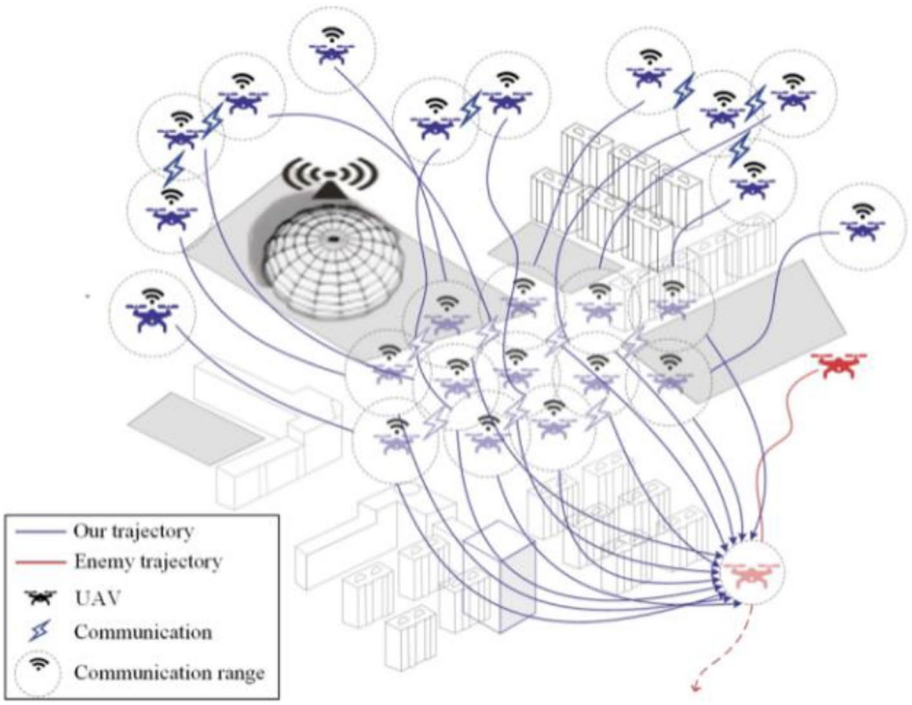

Fig. 11 An example of the decision-making trajectory for collaborative attacks by trained UAVs

for enhancing the operational capabilities of UAS and expanding their application fields. Certainly, our study is not without its limitations. Specifically, it does not thoroughly investigate scenarios with more stringent communication constraints. Furthermore, the challenge of the curse of dimensionality arises when scaling up to manage larger clusters of UAVs. In future research, we will endeavor to address the challenges posed by the increasing number of UAVs and to solve the coordination problem of larger scale UAVs.

Abbreviations

UAV	Unmanned Aerial Vehicle
MARL	Multi-agent reinforcement learning
Dec-POMDPs	Decentralized partially observable Markov decision process
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
LSTM	Long Short-Term Memory
MLP	Multi-layer Perceptron

Acknowledgements
The authors would like to thank all anonymous reviewers for their invaluable comments.

Authors' contributions
Zhang Ting-Ting, Chen Yan and Dong Ren-zhi wrote the main manuscript including experimental design and implementation, Chen Tao provided the requirements analysis, Liu Yan, Zhang Kai-Ge provided the application analysis, Song Ai-Guo, Lan Yu-Shi provide theoretical guidance. All authors reviewed the manuscript.

Funding
This work is supported by China Postdoctoral Science Foundation (No. 2019M651991)

Data Availability
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 11 November 2024 Accepted: 11 February 2025
Published online: 24 February 2025

Reference

1. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L (2017) Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems NY, USA, pp 6000–6010
2. Hv Hasselt, Guez A, Silver D (2016) Deep reinforcement learning with double Q-learning. The Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016). Arizona, USA, Phoenix, pp 2094–2100
3. Chen T, Yang Q, Chen Y (2023) Jump-NERF: an approach to removing glare and pseudo shadows caused by glass in architectural spaces. 2023 China Automation Congress, Chongqing, pp 8365–8370
4. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
5. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81
6. Wu Z, Shen C, van den Hengel A (2019) Wider or deeper: revisiting the ResNet model for visual recognition. Pattern Recogn 90:119–133
7. Zhang T, Chai L, Wang S, Jin J, Liu X, Song A, Lan Y (2022) Improving autonomous behavior strategy learning in an unmanned swarm system through knowledge enhancement. IEEE Trans Reliab 71(2):763–774
8. Zhang T, Lan Y, Song A (2021) A review of autonomous cooperative technologies for unmanned cluster systems. J Command Control 7(2):127–136
9. Zhang Y, Wu F, Wang M, Duan H, Zhang Z, Wang H (2021) Deep reinforcement learning-based autonomous behavioral decision making for unmanned combat vehicles. Fire Control Command Control 46(4):72–77
10. Wang H, Bai H, Li F et al (2024) Throughput maximization for covert UAV relaying system. IEEE Trans Veh Technol 73(3):4429–4434
11. Bai H, Wang H, Du J, He R, Li G, Xu Y (2024) Multi-hop UAV relay covert communication: a multi-agent reinforcement learning approach. In: 2024 International Conference on Ubiquitous Communication (Ucom), Xi'an, China. IEEE, pp 356–360
12. Zhao L, Zhang Y, Yao M, Guo Y (2021) Development and prospect of unmanned aerial vehicle swarm coordination technology. Radio Eng 51(8):823–828
13. Wu H, Li X, Deng Y (2020) Deep learning-driven wireless communication for edge-cloud computing: opportunities and challenges. J Cloud Comput 9(1):1–14
14. Chen Y, Zhou R (2024) Coverage path planning for multi UAV collaborative environment under communication constraints. J Chin Inert Technol 32(3):273–281
15. Xiao D, Jiang J, Zhou J, Yu C (2018) Multi-UAV cooperation search for moving targets under limited communication. J Harbin Eng Univ 39(11):1823–1829
16. Cheng C, Zhang Y, Chu H, Zhao W (2019) Distributed multi-UAV cooperative formation control simulation. Comput Simul 36(5):31–37
17. Cao J, Liu X, Li S, Chu W (2022) Multi-target Mission Planning for Multi-UAV Cooperative Search in Multi-base Systems with Limited Communication. Ordnance Ind Autom 41(11):89–92
18. Yu H, Lin Y, Jia L, Li Q, Zhang Y (2022) Distributed strategy for multi-target rescue with communication-constrained UAV swarms.Chin. J Internet Things 6(3):103–112
19. .Fu X, Pan J, Wang H, Gao X (2020) A formation maintenance and reconstruction method of UAV swarm based on distributed control. Aerosp Sci Technol 104:105981
20. Bramblett L, Bezzo N (2023) Epistemic planning for multi-robot systems in communication-restricted environments. Front Rob AI 10:1149439
21. He G (2002) Destination-sequenced distance vector (DSDV) protocol. Networking Laboratory, Helsinki University of Technology 135:1–9
22. Zhu Y, Zhao D, He H, Ji J (2017) Event-triggered optimal control for partially unknown constrained-input systems via adaptive dynamic programming. IEEE Trans Ind Electron 64(5):4101–4109
23. Wang Y, Yang M, Dong R (2023) Efficient potential-based exploration in reinforcement learning using inverse dynamic bisimulation metric. In: Proceedings of the 37th International Conference on Neural Information Processing Systems, pp 38786–38797
24. Yang M, Dong R, Wang Y, Liu F, Du Y, Zhou M, Leong Hou U (2023) TieComm: learning a hierarchical communication topology based on Tie theory. In: 28th International Conference on Database Systems for Advanced Applications, pp 604–613
25. Yang M, Zhao K, Wang Y, Dong R, Du Y, Liu F, Zhou M, Leong Hou U (2024) Team-wise effective communication in multi-agent reinforcement learning. Auton Agent Multi-Agent Syst 38:1–36
26. Yang Y, Luo R, Li M (2018) Mean field multi-agent reinforcement learning. In: 35th International Conference on Machine Learning, pp 5571–5580
27. Wu F, Zilberstein S, Jennings NR (2013) Monte-Carlo expectation maximization for decentralized POMDPs. In: Proceedings of the 23rd international joint conference on artificial intelligence, pp 397–403

28. Amato C, Chowdhary G, Geramifard A (2013) Decentralized control of partially observable Markov decision processes. In: 52nd IEEE Conference on Decision and Control. IEEE, pp 2398–2405
29. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. 4th International Conference on Learning Representations. San Juan, Puerto Rico, pp 1–14
30. Lowe R, Wu YI, Tamar A (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp 6382–6393
31. Yu C, Velu A, Vinitsky E (2022) The surprising effectiveness of ppo in cooperative multi-agent games. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp 24611–24624
32. Peng H, Shen X (2021) Multi-agent reinforcement learning based resource management in MEC-and UAV-assisted vehicular networks. IEEE J Sel Areas Commun 39(1):131–141
33. Pateria S, Subagdja B, Tan A (2021) Hierarchical reinforcement learning: a comprehensive survey. ACM-CSUR 54(5):1–35

Publisher's Note