IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 August 2023, accepted 24 September 2023, date of publication 28 September 2023, date of current version 5 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320137

RESEARCH ARTICLE

A Sequential Decision Algorithm of Reinforcement Learning for Composite Action Space

YUAN GAO[1,2], **YE WANG**[1,2], **LEI ZHANG**[1,2], **LIHONG GUO**[1], **JIANG LI**[1], **AND SHOUHONG SUN**[1]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]School of Optoelectronics, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Shouhong Sun (sunsh@ciomp.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61977059.

ABSTRACT It is the key research object of electronic warfare to use UAV(Unmanned Aerial Vehicle) clusters to carry out electronic countermeasure tasks. The UAV carries loads such as reconnaissance and interference at the same time, which makes it necessary to simultaneously decide multiple types of actions—namely, compound actions—which poses a challenge to intelligent decision-making algorithms. Considering the problem of action-space dimensional complexity and weak collaboration between decisions in multi-agent scenarios with composite actions, this study proposed a decision algorithm involving a multi-agent reinforcement-learning sequence, which combined joint composite actions into sequential decision, reducing the difficulty of a single decision and enhancing the collaboration between various agents and their individual decisions. Because long decision sequences required better depth modeling and had high variance, a DeLighT module was added to the naïve transformer model to increase the depth and baseline techniques, which were used to reduce the variance in the value estimation. The simulated results verified the effectiveness of the proposed algorithm in the UAV cooperative combat scenario, where each agent had a composite action space and showed better performance than the existing algorithms.

INDEX TERMS Baseline, composite action space, multi-agent reinforcement learning, transformer, UAVs sensing and jamming strategy.

I. INTRODUCTION

In modern warfare, in addition to the four traditional combat domains of land, sea, air and sky, the electromagnetic spectrum has also become an important combat domain. Electronic warfare (EW) is the science and art of depriving the enemy of the ability to use the electromagnetic spectrum while protecting one's own access. The definition in the *Soviet Military Encyclopedia*, issued by the Russian government, was that electronic warfare referred to the process of using electronic means to attack other enemy assets to affect the state of the combat environment, and the goal of electronic warfare was to reduce the operational effectiveness of enemy forces (including command and control capabilities and weapon system application capabilities).

General Rastochkin, Commander of the Russian Electronic Warfare Force, pointed out that the use of a unified miniaturized reconnaissance-and-jamming module mounted on a drone to suppress the radio in local areas in order to generate controllable areas was the key research object of electronic warfare development. At present, the research of electronic warfare strategy has been focused on the sub-domains of intelligent perception, interference resource allocation, cognitive interference decision-making, and anti-jamming decision-making, but there have been few studies on complex decision-making for the comprehensive electronic warfare observe–orient–decide–act (OODA) loop.

Reinforcement learning is a method used to describe and solve a problem for which agents attempt to maximize returns or achieve specific goals by applying learning strategies to their environmental interactions. It is a strategy for learning

The associate editor coordinating the review of this manuscript and approving it for publication was Kah Phooi (Jasmine) Seng.

specific tasks and can also be employed to mimic hypothetical enemy behaviors to assist electronic warfare experts in various stages of electronic warfare.

In this study, the UAV swarm cooperative escort task was used as the model environment, and the multi-agent reinforcement learning algorithm was used to explore the strategy while the UAV was used as the agent. The agent had both cognitive reconnaissance and cognitive jamming capabilities, and could complete the entire OODA loop. Furthermore, the agents needed to make decisions on flight, reconnaissance, interference, and other actions at the same time, which challenged the proposed multi-agent decision-making strategy as an exploration method.

A. RELATED WORK

First of all, our target scenario, multi-UAV cooperative escort mission, belongs to the field of electronic warfare. A game scenario exemplifies the struggle between two opposing sides in electronic warfare, which becomes a dynamic process of mutual recognition and mutual avoidance. Only when the electronic warfare equipment has the ability of learning during the confrontation and can adjust its own response strategy efficiently through the identification of the opponent feedback state in order to understand the initiative behind future electronic warfare. Therefore, the concept of a cognitive electronic warfare system with a learning capacity was proposed for both domestic and international application [1].

The definition of cognitive electronic warfare is based on electronic warfare equipment with cognitive abilities and is focused on electronic warfare operations that use autonomous interactive learning capabilities in electromagnetic environments and dynamic intelligent confrontation capabilities. It includes three basic elements: cognitive reconnaissance, cognitive interference, and cognitive defense [2].

Our research focuses on the combat strategy of multi-UAV electronic warfare. The research of electronic warfare equipment technology [3] determines the lower limit of electronic warfare capability, while the research of electronic warfare strategy concerned in this paper determines the upper limit of electronic warfare capability.

At present, the research focus of electronic warfare strategy has been divided into the following tasks: intelligent electromagnetic sensing, jamming resource allocation, cognitive jamming decision-making and cognitive anti-jamming decision-making. The research concerning intelligent-sensing tasks has focused on sensing systems [4], [5], weak signals [6], and cooperative sensing [7]. The task of jamming resource allocation in electronic warfare must consider how to obtain the best overall electronic attack with limited overall jamming resources, making this a typical combinatorial optimization problem [8], [9]. It also involves the task reassignment problem in dynamic environment [10]. Cognitive jamming decision-making involves the intelligent development of electronic jamming [11], [12], [13] while cognitive anti-jamming is the corresponding intelligent defense strategy [14], [15], which has included some anti-jamming performance evaluation models [16], [17].

The general process of electronic warfare is to first intelligently perceive the battlefield; determine the target for interference, according to the perception results; assign tasks to the existing jamming resources; and make decisions that subsequently suppress the enemy in order to complete the OODA loop. Similarly, the interfered party could also implement anti-jamming strategies. At present, the research has focused on the aforementioned sub-areas but has lacked exploration and research pertaining to the complete electronic warfare process, much less the integration of each part to form a complete electronic warfare scenario. That's what our research focuses on. The training environment of the subject was an electronic countermeasure model with a UAV cluster that included several cooperative tasks, including control, reconnaissance and jamming task, ensuring is a comprehensive electronic warfare scene.

The UAV swarm, the agents in our scenario, operation is an important element in future war scenarios. Many scholars have also conducted a lot of research on the strategy of UAV cluster task completion. Swarm intelligence is an important method to solve the multi-UAV task [18]. The quality of the control strategy directly affected the safety and the stability of the UAV swarm when completing the task [19]. Duan et al. studied a wolf pack's intelligent behavior and applied it to a cooperative UAV swarm in a decision-making simulation [20]. Inspired by bird behavior, Shen and Wei designed a hierarchical cluster control framework [21]. Besides, heuristic algorithms are an option. Gao and Li proposed a distributed cooperation method and information processing for a UAV swarm based on situational awareness and a consensus to strengthen adaptability in a complex environment [22]. To address a cooperative search problem, Zhang et al. used an improved particle swarm optimization algorithm to allocate the UAV reconnaissance area and maximize the utility of the UAV cluster [23]. Of course, reinforcement learning is also a good option. Yue et al. proposed a secure transfer soft-AC algorithm with security constraints for maximizing revenue [24]. Baek and York adopted distributed algorithms so UAV swarms could track ground targets and proposed an optimal sensor management technology and consensus-based decision algorithm to minimize the uncertainty of the target location [25].

Our target scenario requires multiple UAVs to make flight, reconnaissance, interference and other actions at the same time, and the task is relatively complex. We choose multi-agent reinforcement learning algorithm based on Markov decision process, which is more suitable for solving this complex decision task.

At present, multi-agent reinforcement learning has been applied to complex problems, such as traffic control [26], multi-loop chemical process control [27], pathogenic gene prediction [28], and military operations [29]. It has been a powerful tool for studying complex task decision-making. Ahmed studied the effective interaction system between agents [30]. The information obtained in the real world was not necessarily accurate. Therefore, to solve this problem, Chen et al. proposed multi-agent fault-tolerant reinforcement learning [31], whereas Riley et al. proposed a secure multi-agent reinforcement learning method for security-critical and task-sensitive scenarios [32]. Malysheva et al. proposed the MAGNet method and introduced the concept of an environmental correlation graph in reinforcement learning [33]. Liu and Tan proposed feudal latent space exploration for multi-agent reinforcement learning and guided a coordinated exploration using multiple agents by learning the latent structure [34]. Kim et al. analyzed the problem of the collaboration paradox caused by ‘‘lazy’’ agents [35]. Kuba et al. performed a rigorous mathematical analysis of the high variance in the estimations of the policy gradient method and derived the optimal baseline to achieve the minimum variance [36]. Chen et al. introduced the concept of a hierarchical attention map to enhance the scalability of the agent model [37].

In the setting of our target scene, the agent needs to decide on multiple actions at the same time, which also causes the problem of dimensional explosion in the action space. We want to alleviate the problem of dimensional explosion by treating multiple actions as an action sequence and making decisions in turn. The most successful model for processing squential data is Transformer.

The Transformer has achieved great success in many artificial intelligence fields, such as natural language processing, computer vision, and audio processing. It has become the most widely used and popular sequential model among network architectures and has attracted the attention of scholars in the field of reinforcement learning. Chen et al. abstractly described reinforcement learning as a sequential modeling problem, combining transformer framework with offline reinforcement learning [38]. Zheng et al. proposed an online decision transformer algorithm based on sequential modeling, which combined offline pre-training and online fine-tuning paradigms into a unified framework [39], which was also applicable to multi-agent systems [40]. Finally, Wen et al. proposed a training framework for cooperative multi-agent scenarios based on a transformer framework [41].

We find that Transformer in the field of multi-agent reinforcement learning is generally used to solve the problem of dimensional explosion of joint action space caused by the number of agents, that the problem of multiple agent synchronous decision-making is described as a sequential decision-making problem, and the cooperation between agents is enhanced while the decision-making dimension is reduced. However, the problem of composite action space dimension explosion caused by multiple actions of single agent is rarely mentioned, which is also the difficulty of strategy exploration of multi-UAV cooperative electronic countermeasures task designed by us.

B. MOTIVATIONS AND CONTRIBUTIONS

In the process of strategy learning for the target scenario, multi-UAV cooperative electronic countermeasures escort mission, we encountered the following problems:

Firstly, the increase in the number of simultaneous decisions of a single agent has expanded the composite action space and increased the difficulty of the strategy to converge optimally. Secondly, the increase from a single agent to multiple agents has introduced higher performance requirements for the decision model. Therefore, the algorithm adopted in this paper is an actor–critic framework. The critic part estimates the value of the current action and then updates the actor. We use the Monte Carlo (MC) method to estimate the value. However, the dimensional dynamics increases the uncertainty of the action, resulting in high variances when using the MC method. Our study makes the following contributions:

1) In order to alleviate the dimensional complexity caused by the compound action space, the decision process of each step is divided into a decision sequence, and only one action is selected for each decision, which reduces the dimension of each decision. The transformer model is used to process the decision sequence and ensure that there is a correlation between the decision results in the sequence, that is, the collaboration between actions.
2) In order to alleviate the problem of poor convergence, the DeLighT module is added to the naïve transformer model to improve the depth of the network and enhance its performance.
3) In order to alleviate the high variance of the policy gradient and Monte Carlo methods, an independent critic network is established by using a COMA (Counterfactual Multi-Agent (policy gradients)) network to obtain the values of all the next optional actions of an agent in a certain state, and the baseline is calculated.
4) In this study, a series of comparative experiments were carried out to verify the effectiveness of the MA2DBT algorithm in the complex action space, and comparative ablation experiments evaluated the cooperative escort mission assigned to the UAV swarm, and the contribution of each part of the algorithm was explored.

The rest of this article is organized as follows. In Section II, the system model of UAV swarm and their cooperative escort mission is described in detail. Section III presents the MA2DBT algorithm for a series of effects caused by the composite actions. In Section IV, the simulation results are discussed and compared to other multi-agent reinforcement learning methods. Finally, Section V summarizes the research work.

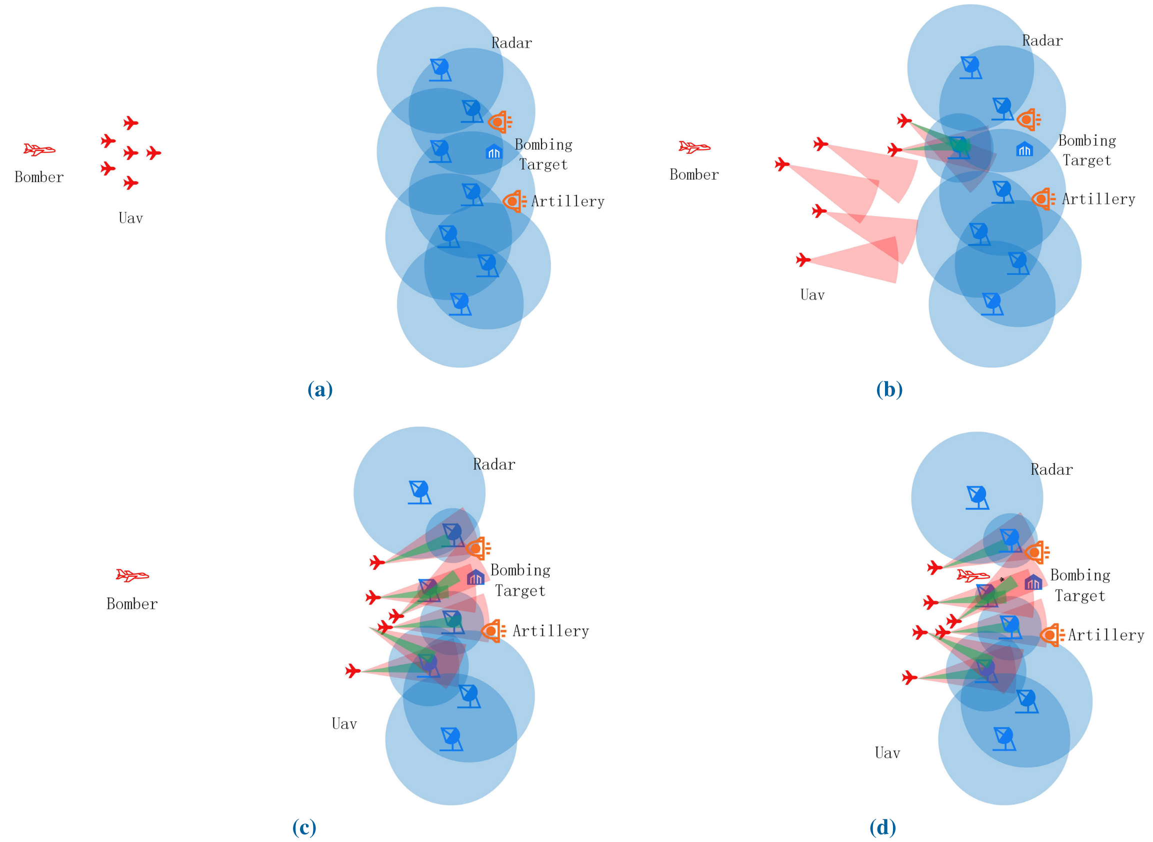


FIGURE 1. UAV swarm escort mission. From (a) to (d) was the ideal electronic warfare process.

II. SYSTEM MODEL

We studied strategic exploration in a scenario of UAV swarms performing escort missions. In this scenario, the radar cluster is the blue and the UAV cluster had to weaken the blue reconnaissance capabilities and avoid being damaged by blue firepower through electronic countermeasures against the blue reconnaissance and firepower strike coverage in order to escort a bomber within feasible attack distance so it could destroy the target.

The aforementioned environment could be modeled as decentralized partially observable Markov decision processes (Dec-POMDPs) $< \mathcal{N}, \boldsymbol{O}, \boldsymbol{A}, \mathcal{R}, \mathcal{P}, \gamma >$. In this scenario, $\mathcal{N} = 1, \ldots, n$ is the set of agents; $\boldsymbol{O} = \prod_{i=1}^{n} O^i$ is the joint observation space, that is, the combination of all agent observation spaces; $\boldsymbol{A} = \prod_{i=1}^{n} A^i$ is the joint action space, that is, the combination of all agent action spaces; $R : \boldsymbol{O} \times \boldsymbol{A} \rightarrow [R_{min}, R_{max}]$ is the joint reward function of the agents; $P : \boldsymbol{O} \times \boldsymbol{A} \times \boldsymbol{O} \rightarrow R$ is the probability function of the Markov decision process's state transition; and $\gamma \in [0, 1)$ is the decay factor.

In a time-step, $t \in \mathbb{N}$, agent $i \in \mathcal{N}$ selects action a_t^i from its action space based on observation $o_t^i \in O$ obtained from the current environment and the current strategy π_i. At each time-step, all agents make decisions simultaneously based on their own observations, without sequential dependencies. At the end of each time-step, the entire team receives a team reward $R(\boldsymbol{o}_t, \boldsymbol{a}_t)(\boldsymbol{o} = o^1, \ldots, o^n, \boldsymbol{a} = a^1, \ldots, a^n)$ and the state of the next time-step state s_{t+1}(or observation o_{t+1}). After infinite execution of this process, the agent obtains cumulative discounted rewards $R^{\gamma} = \sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{o}_t, \boldsymbol{a}_t)$.

The escort mission of the UAV swarm is a multi-agent cooperative mission with compound action space. The UAV swarm was composed of six UAVs that were integrated with reconnaissance and jamming. Each UAV was considered an agent. Each agent made decisions on its flight direction, flight speed, reconnaissance direction, jamming direction, jamming frequency, jamming intensity, etc., to realize the detection, positioning, and interference of the blue radar, weaken the radar reconnaissance, and complete the bomber-escort task (the bomber was not an agent). As a non-player character (NPC), the blue side was equipped with seven omnidirectional reconnaissance radars, and at least three radars covered the bombing position. In addition, the blue side had two firepower points that were distributed near the target. The firepower launch depended on the guidance of the radar, so the first priority of the agent was to ensure that it was not detected by the radar, so as to ensure that it would not be incapacitated.

In this scenario, the reward of the agent was divided into three parts, the flight reward, the reconnaissance reward, and the jamming reward. The agent needed to be close to the reconnaissance distance to detect the blue radar, and the blue radar could experience interference by the drone within the interference range, so the flight reward would be given according to the distance between the drone cluster and the radar cluster. The reconnaissance reward could be obtained for every reconnaissance, capture, and location of a radar. The interference of the agent with the radar affected the radar's reconnaissance ability. The interference reward was calculated according to the degree of weakening and upon the arrival of the bomber to the safe bombing position.

The general process of the whole scene was as follows,as shown Figure 1. First, the scene is initialized, including seven UAVs (agents) and one bomber (NPC) on the red side, whose goal is to open up a safe area to help bombers complete their strike tasks through electronic countermeasures, while the blue side includes seven radars and two fire units guarding near the target, as shown in Figure 1(a). Then, the UAV conducts reconnaissance on the radar, and interferes with the detected radar that affects the bomber's striking the target to suppress its reconnaissance, as shown in Figure 1(b). When the UAV detects all the radars that cover the attack point within the reconnaissance range, the UAV interferes with the threatening radar to suppress its detection and open up a safe path for the bomber, as shown in Figure 1(c). Finally, the bombers arrive at the safe attack position to complete the bombing (this step is an established strategy), as shown in Figure 1(d). So far, the multi-UAV cooperative escort task has been completed. Throughout the mission, drones will also face the risk of being detected by radar and hit.

Next, the key calculation model in the scene design is expounded. Firstly, the UAVs needed to detect the position of the blue radar. The UAVs adopted a directional reconnaissance mode, which had a long reconnaissance distance but a narrow range. When the UAV detected the radar, the UAV cluster would receive a reconnaissance result, which was regarded as a capture of the radar information. The radar information was captured more than three times to complete the determination of the radar frequency band. Two or more reconnaissance results could be cross-referenced to accurately locate the radar position.

Red drone reconnaissance blue radar follows the following equation:

Let the probability of finding the target in the i-th observation be $q_i (i = 1, 2, \ldots, n)$ and let the random variable X denote the number of observations (serial numbers) of the first discovery of the target. Therefore, the probability that the target would not be found in the first $n-1$ observations but found in the n-th observation is

$$P(X = n) = q_n \prod_{i=1}^{n-1} (1 - q_i) \tag{1}$$

Therefore, we found the expected number of target observations:

$$E(X) = \sum_{n=1}^{\infty} nP(X = n) \tag{2}$$

The variance is expressed by the following:

$$\begin{aligned}\sigma_X^2 &= E\left\{[X - E(X)]^2\right\} = E\left(X^2\right) - E(X)^2 \\ &= \sum_{n=1}^{\infty} n^2 P(X = n) - E(X)^2 \end{aligned}\tag{3}$$

If q_i can be seen as a constant $q_i = q, i = 1, 2, \ldots, n$, that is, when the conditions do not change greatly during the search, we have

$$P(X = n) = q(1 - q)^{n-1} \tag{4}$$

$$E(X) = \sum_{n=1}^{\infty} nq(1 - q)^{n-1} = \frac{1}{q} \tag{5}$$

$$\sigma_X = \sqrt{\sum_{n=1}^{\infty} n^2 q(1 - q)^{n-1} - \frac{1}{q^2}} = \frac{\sqrt{1 - q}}{q} \tag{6}$$

The random variable X obeys a geometric distribution.

Meanwhile, the blue radar is also detecting the position of the red UAV, and the radar detection of the UAV will be affected by the jammer.

The detection probability of the radar to detect the UAVs is calculated according to the following formula:

$$P_{d_i} = \int_0^{\infty} e^{-t} \left\{1 - \varphi\left[\frac{Y_0 - n_0\left(1 + S_{n_i} t\right)}{\sqrt{n_0\left(1 + 2S_{n_i} t\right)}}\right]\right\} dt \tag{7}$$

Under interference conditions, the radar detection probability $\left(P_{d_i}\right)$ depends on the ratio of the signal energy and interference energy, where function $\varphi(x)$ is

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt \tag{8}$$

where, S_{n_i} is the signal-to-interference ratio of a single pulse in the i-th contact with the target, is calculated as follows:

$$S_{n_i} = \frac{P_t G_t^2 \Delta f_i \sigma}{4\pi P_i G_i \gamma_i G_i(\theta) \Delta f_r L} \cdot \frac{R_i^2}{R_t^4} \tag{9}$$

$G_i(\theta)$ is the gain of the radar antenna in the direction of the jammer; therefore,

$$f(x) = \begin{cases} G_t, & 0 \leq |\theta| \leq \dfrac{\theta_{0.5}}{2} \\ k\left(\dfrac{\theta_{0.5}}{\theta}\right)^2 G_t, & \dfrac{\theta_{0.5}}{2} < |\theta| \leq 90^\circ \\ k\left(\dfrac{\theta_{0.5}}{90}\right)^2 G_t, & 90^\circ < |\theta| \leq 180^\circ \end{cases} \tag{10}$$

where k is constant.

III. MULTI-AGENT MULTI-ACTION DELIGHT BASELINE TRANSFORMER

In a multi-agent collaboration task, the collaboration between agents is an important consideration for the decision algorithm to achieve better performance. When the action space of an agent is a composite action space formed by the Cartesian product of multiple types of actions, the collaboration between different types of actions also guides the algorithm to obtain an optimal strategy. For an scenario with composite action space composed of multiple types of actions (this study focuses on a discrete action space),such as our scenario that multiple UAVs need to perform reconnaissance, jamming and other actions at the same time to complete the escort mission, in our experiment,the existing algorithms for reinforcement-learning-based strategic exploration, such as PPO (Proximal Policy Optimization) and DDPG (Deep Deterministic Policy Gradient), no matter it is single agent task or multi-agent task, The strategy obtained is not ideal for the completion of the task. For this phenomenon, we analyze that multiple types of actions lead to the explosive expansion of the agent's action space, resulting in aimless exploration and ignoring the influence of cooperation between different types of actions on decision making. Based on this, we propose Multi-Agent Multi-Action DeLighT Baseline Transformer (MA2DBT) algorithm for decision tasks in a composite action space.

A. SPLITTING COMPOSITE ACTIONS FOR SEQUENTIAL DECISION-MAKING

In multi-agent collaboration tasks, sequential decision-making methods have been used for collaborative decisions among agents, that is, to determine their own decisions based on their individual observations as well as the decisions of previous agents. This multi-agent-ordered decision streamlined the update of their joint strategies. According to Theorem 1, maximizing each agent's local advantage is equivalent to maximizing their joint advantage.

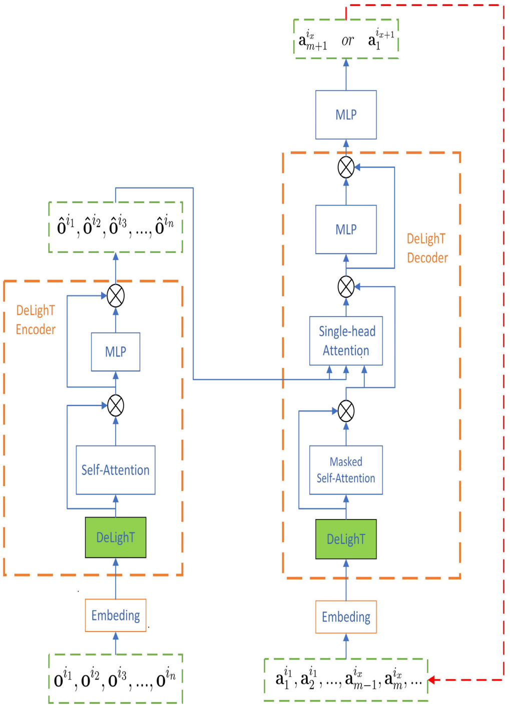
Theorem 1. *(multi-agent advantage decomposition): Let $i_{1:n}$ be a permutation of agents. Then, for any joint observation $o \in O$ and joint action $a = a^{i_{1:n}} \in A$, the following equation always hold with no further assumptions needed [36]:*

$$A_{\pi}^{i}\left(o, a^{i_{1:n}}\right) = \sum_{m=1}^{n} A_{\pi}^{i_m}\left(o, a^{i_{1:m-1}}, a^{i_m}\right) \tag{11}$$

Similarly, for a decision-making task in a compound action space composed of multiple types of actions, the cooperative nature of the actions can also be enhanced by sequential decision-making.

Corollary 1. *(multi-action advantage decomposition): Let i be any agent and $a_{1:n}$ be a permutation of actions. Then, for any observation o and joint action $a = a_{1:n} \in A$, the following equation always holds:*

$$A_{\pi}^{i}\left(o, a_{1:n}\right) = \sum_{m=1}^{n} A_{\pi}^{i}\left(o, a_{1:m-1}, a_m\right) \tag{12}$$



$a_{m+1}^{i_x}$ or $a_1^{i_{x+1}}$
MLP
MLP
DeLighT Decoder
Single-head Attention
Masked Self-Attention
DeLighT
Embeding
$a_1^{i_1}, a_2^{i_1}, ..., a_{m-1}^{i_x}, a_m^{i_x}, ...$
$\hat{o}^{i_1}, \hat{o}^{i_2}, \hat{o}^{i_3}, ..., \hat{o}^{i_n}$
DeLighT Encoder
MLP
Self-Attention
DeLighT
Embeding
$o^{i_1}, o^{i_2}, o^{i_3}, ..., o^{i_n}$

FIGURE 2. The Actor architecture of MA2DBT.

Further, We extended Corollary 1 to multi-agent scenarios, where multiple agents have multiple types of actions from which to choose; therefore,

Corollary 2. *(multi-agent multi-action advantage decomposition): Let $i_{1:t}$ be a permutation of agents and $a_{1:n}$ be a permutation of actions. Then, for any observation o and joint composite action $a = a_{1:n}^{1:t} \in A$, the following equation always holds:*

$$A_{\pi}^{i}\left(o, a_{1:n}^{i_{1:t}}\right) = \sum_{m=1,b=1}^{n,t} A_{\pi}^{i_b}\left(o, a_{1:m-1}^{i_b}, a_m^{i_b}\right) \tag{13}$$

The theorem and corollary provide a concept for the decision-making task with compound action space. Suppose, in any state, agent i chooses action $a_b^i, A_{\pi}^{i_b}\left(o, a_b^i\right) > 0$, with positive advantage; then, assume that all the proceeding agents know the choice a_b^i. Based on this premise, each subsequent step tends towards the action with positive advantage, that is, $A_{\pi}^{i_b}\left(o, a_i^{i_b}, a_i^{i_b+1}\right) > 0$. The theorem and corollary ensure the positive advantage of the joint composite action. In addition, each decision only needs to be selected in the action space of the agent. The complexity of the overall exploration process is additive rather than multiplicative. It does not need to consider the composite actions of different types in the selection step, which reduces the difficulty of the strategic exploration while ensuring collaboration.

B. TRANSFORMER AND DELIGHT

The decision-making process of multiple agents with compound action space is decomposed into a sequential process of each action of an agent. The process of sequential decision-making is remarkably similar to that of natural language processing. For example, in the field of machine translation, the model obtains the first word of the translation results according to the first word of the sentence to be translated, so it could then output the second word of the translation based on the first word of the translation results, taking the second word of the translated sentence the input, and so on, until the whole sentence is translated. There is a correlation between the translation results. In the field of intelligent decision-making, this phenomenon is analogous to collaborative decision-making.

The transformer model uses an encoder–decoder architecture, and its encoder and decoder are not traditional RNN (Recurrent Neural Network) structures. Instead, they are individual encoder and decoder stacks comprising consecutive encoders and decoders with the same structure. Each encoder contains a self-attention module and a feed-forward neural network. Each sub-network has a residual link, and the results are normalized after each residual synthesis. In addition to the self-attention mechanism and fully connected feed-forward neural network, and similar to the encoder, an attention module is added between the two sub-networks in each decoder. The three sub-networks all have residual links and are normalized after residual synthesis. The residual connections can prevent network degradation caused by gradient disappearance and improve the network depth.

The transformer establishes the relationships in the data based on the attention mechanism: the higher the value, the higher the correlation between the two. The transformer does not use an RNN structure, so parallel computing can be performed, which greatly improves its operational speed.

The observations of all the agents are used as sentences to be translated, and all types of actions determined by all the agents are used as the translation results for serialization. Due to the complexity of the task and the action space, we need to further improve the performance of the sequential model. The common practice is to expand the model depth, typically by increasing the number of hidden layers or stacking additional transformer blocks, but these practices increase the difficulty of the model training.

Therefore, we designed a network model based on the DeLighT model, which had a deeper-yet-lighter transformer structure. The DeLighT model allocates the model parameters more efficiently. It uses DExTra transform in each transformer block and uses block-level scaling at the input end to make the DeLighT block shallower and narrower while allowing the DeLighT block at the output end to be deeper and wider. This approach deepens the network model while maintaining a smaller number of parameters.

As shown in Figure 2.The encoder is responsible for encoding the input. An observational sequence $(o^{i_1}, \ldots, o^{i_n})$ could enter in any order. After passing through the encoder's DeLighT module, self-attention module, and feed-forward neural network, the encoding result is obtained and then transmitted to the attention module in the decoder. The decoder is responsible for obtaining the action sequence, which is composed of an initial action (similar to the initial identification of the text sequence) as well as a decoder composed of a DeLighT module, a self-attention module, a masked–attention module, and a feed-forward neural network. It calculates the next sequence of actions and then continuously loops this process until the entire action sequence is determined. The input of the attention module also includes the encoding result of the encoder and the residual input of the sub-attention module. The role of the mask is to ensure that the decision of the current action is not affected by future data.

The loss function used to train the network was the following:

$$L_{Actor}(\theta) = \frac{1}{Tnd}\sum_{m=1}^{n}\sum_{b=1}^{d}\sum_{t=0}^{T-1}\min(r_t^{i_m^b}(\theta)\hat{X}_t, clip(r_t^{i_m^b}(\theta), 1 \pm \epsilon)\hat{X}_t) \tag{14}$$

where, $r_t^{i_m^b}(\theta) = \frac{\pi_\theta^{i_m^b}(a_t^{i_m^b}|\hat{\boldsymbol{o}}_t^{i_{1:n}}, \hat{\boldsymbol{a}}_t^{i_{1:m-1}^{1:b-1}})}{\pi_{\theta_{old}}^{i_m^b}(a_t^{i_m^b}|\hat{\boldsymbol{o}}_t^{i_{1:n}}, \hat{\boldsymbol{a}}_t^{i_{1:m-1}^{1:b-1}})}$

and where n is the number of agents, d is the number of action types of one agent, T is the episode length, and $\hat{X}$ is the value advantage function calculated by the critic.

C. CRITIC AND BASELINE

In the multi-agent collaboration task, the agent who makes the greatest contribution and the agent who makes the least contribution receive the same reward. This leads to some agents becoming ''lazy''. The distribution of the agent reward is unfair and should follow the principle of ''more work for more gain''. This is a typical credit assignment problem in multi-agent cooperative tasks.

Our algorithm uses the CTDE (Centralized Training with Decentralized Execution) framework. The transformer model is used as the actor to obtain the action. We observed that the convergence of the actor strategy depends on the estimation of the $\hat{X}$ advantage function, which is, in turn, based on the critic estimation of the value function. For this study, we adopted the Monte Carlo estimation method, which must wait for the end of the current episode to learn. It has the characteristic of unbiased estimation but with high variance. Therefore, we attempted to reduce the variance of the value function estimation by control-variate subtraction, also known as a baseline track. At the same time, the problem of belief distribution can also make the value estimation of different actions of different agents vary by adding baselines, making the return distribution more reasonable.

Baseline b is a function that does not depend on action a, therefore,

$$E\left[b \cdot \nabla_\theta log\pi_\theta\left(a|s\right)\right] = 0 \tag{15}$$

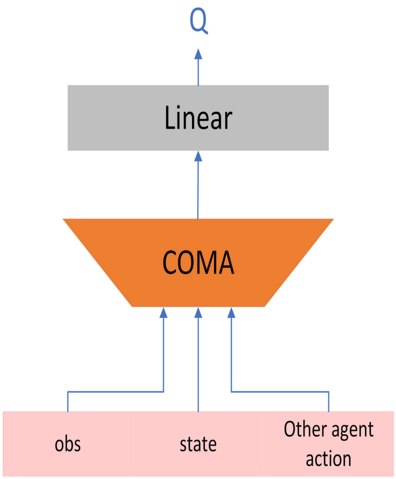


FIGURE 3. The Critic architecture of MA2DBT.

For the calculation of the multi-agent strategy gradient, we calculated the following:

$$\nabla_{\theta^i}\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{s_{0:\infty}\sim d_{\boldsymbol{\theta}}^{0:\infty}, \mathbf{a}_{0:\infty}^{-i}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}^{-i}, a_{0:\infty}^{i}\sim\pi_{\boldsymbol{\theta}}^{i}} \left[\sum_{t=0}^{\infty}\gamma^t Q_{\boldsymbol{\theta}}\left(s_t, \mathbf{a}_t^{-i}, a_t^i\right)\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(a_t^i \mid s_t\right)\right] \tag{16}$$

When we brought in the baseline, we found as shown as Equation (17) at the bottom of the page.

It can be seen that whatever the function b is, the expectation of the gradient is constant, so the addition of the baseline does not affect the correctness of the policy gradient.

In this study, three baselines were applied for different task selections: state value function, counterfactual baseline, and optimal baseline as obtained by mathematical calculation.

First, the state value function was used as the baseline. In this case, the advantage function is the general advantage function.

$$\hat{X} = Q(s, a) - V(s) \tag{18}$$

Second, we used counterfactual baselines. The concept of a counterfactual baseline is to evaluate the contribution of an agent action. We can replace the agent action with a default action and then focus on the change in the team results before and after the replacement. When the results increase, the effect of the current action is not as good as the baseline action. In order to reduce the computational complexity, the average effect of the current strategy was used as the effect of the default action [42].

$$\hat{X} = Q\left(s, a^i\right) - \sum \pi^i\left(a_t|s_t\right) Q\left(s_t, a_t\right) \tag{19}$$

Third, the optimal baseline was obtained by mathematical calculations. In the optimal baseline theory, the causes of variances in a multi-agent environment are analyzed. The estimated variance mainly comes from the variance of the state, the variance of the current agent action, and the variance of the remaining agents [36], as shown as Equation (20), at the bottom of the next page.

The baseline function is affected by the states and the actions of other agents but does not depend on the action of the current agent; therefore, minimizing the third variance minimizes the entire variance.

The optimal baseline was determined by the following:

$$b^{\text{optimal}}\left(s, \mathbf{a}^{-i}\right) = \frac{\mathbb{E}_{a^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\hat{Q}\left(s, \mathbf{a}^{-i}, a^i\right)\left\|\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(a^i \mid s\right)\right\|^2\right]}{\mathbb{E}_{a^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\left\|\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(a^i \mid s\right)\right\|^2\right]} \tag{21}$$

Among the three baselines, the action value function $Q(s, a)$ and the state value function $V(s)$ must be used. Therefore, the algorithm establishes a corresponding evaluation network. On the one hand, the encoder in the transformer was used to calculate the state value $V(s)$ with the coded state of the agent as input and the state value as output for the first baseline. On the other hand, the calculation of the latter two baselines depends on the action value of all the possible actions of an agent at a certain moment. In order to facilitate the calculations,we established a COMA network to estimate the value of the next-possible action of an agent in a certain state. The model structure of the critic is provided in Figure 3. The network takes the observation of an agent, the state of the environment, and all the actions of the remaining agents, as input, and it outputs the value Q of all the next-possible actions of the agent in this state, for the subsequent baseline calculation. A target network was added to the training to ensure the stability of the estimation.

$$\begin{aligned}
\nabla_{\theta^i}\mathcal{J}(\boldsymbol{\theta}, b) &= \mathbb{E}_{s\sim d_{\boldsymbol{\theta}}, \mathbf{a}_{0:\infty}^{-i}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}^{-i}, a_{0:\infty}^{i}\sim\pi_{\boldsymbol{\theta}}^{i}}\left[\sum_{t=0}^{\infty}\gamma^t\left(Q_{\boldsymbol{\theta}}\left(s_t, \mathbf{a}_t^{-i}, a_t^i\right) - b(s_t, \mathbf{a}_t^{-i})\right)\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(a_t^i \mid s_t\right)\right] \\
&= \mathbb{E}_{s_{0:\infty}\sim d_{\boldsymbol{\theta}}^{0:\infty}, \mathbf{a}_{0:\infty}^{-i}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}^{-i}, a_{0:\infty}^{i}\sim\pi_{\boldsymbol{\theta}}^{i}}\left[\sum_{t=0}^{\infty}\gamma^t Q_{\boldsymbol{\theta}}\left(s_t, \mathbf{a}_t^{-i}, a_t^i\right)\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(a_t^i \mid s_t\right)\right] \\
&\quad - \sum_{t=0}^{\infty}\mathbb{E}_{s_{0:\infty}\sim d_{\boldsymbol{\theta}}^{0:\infty}, \mathbf{a}_{0:\infty}^{-i}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}^{-i},}\left[b(s, \mathbf{a}^{-i})\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(a_t^i \mid s_t\right)\right] \\
&= \nabla_{\theta^i}\mathcal{J}(\boldsymbol{\theta})
\end{aligned} \tag{17}$$

The loss function used to train the network was determined by the following:

$$L_{Critic}(\phi) = r + \gamma \sum_{i=0}^{n} \omega_i \hat{Q}_i - \sum_{i=0}^{n} \omega_i Q_i \tag{22}$$

where $\hat{Q}$ is the result of target network and ω is the probability of all actions selected by the agents in a certain state.

According to the above three parts, the MA2DBT algorithm proposed in this paper is composed. The algorithm adopts the classical AC architecture, the Actor makes the decision of the action, while the Critic evaluates the value of the current status and actions, and the evaluation results are used to guide the gradient update of the Actor. The Actor network adopts the structure of Encoder-Decoder, and the observation information is encoded by the Encoder. The Decoder makes sequence decision according to the observation information until a complete action sequence is decided, while the Critic network adopts the structure as shown in Figure 3 to obtain the action value function. The baseline method is used to reduce the variance of the value estimate, so as to guide the update of actors more smoothly. The detailed algorithm flow is as Algorithm III-C(on the next page).

IV. EXPERIMENTS

In order to evaluate whether the MA2DBT algorithm effectively improved the decision-making ability of the multi-agent algorithm for the UAV-swarm escort mission in a composite action space, we compared it to some current state-of-the-art algorithms. In order to obtain a more universal conclusion, we carry out experiments in a stepped way to compare the performance of the algorithm, from single-agent tasks to multi-agent tasks with simple tasks, finally, to the multi-agent task of complex tasks. In the second part, the multi-UAV cooperative electronic countermeasures task scenario built by our research group is adopted. It is also multi-agent task, but the electronic countermeasures task can be subdivided into reconnaissance and jamming tasks, both of which need to consider the position of the UAV, and the tasks are relatively complex. In addition, we also conducted some ablation experiments to explore some properties of our algorithm. First of all, our algorithm is a sequential decision-making method, so it is necessary to explore the influence of decision order on the results. Secondly, scholars have proposed a variety of baseline methods, and we have explored the effects of different baselines in different scenarios.

A. COMPARATIVE EXPERIMENT: THE EFFECT OF SPLITTING COMPOSITE ACTION SPACE

1) SINGLE AGENTS WITH COMPOSITE ACTIONS SPACE

Firstly, several popular reinforcement-learning algorithms were compared in a compound action space with a single-agent task. Here, we used a single soccer agent scenario in Google Research Football, and in order to ensure the similarity of the situation, we modified the action space from each agent only deciding one step to one where each decision consisted of multiple future steps: the more steps of a decision, the more complex the action space.

In the Figure 4, We observed that the convergence rate of the MA2DBT algorithm was better than that of MAT and MAPPO for two and three steps in the future. As the complexity of the action space increased, the convergence rate of the algorithm decreased. Since the composite action space of each agent reached 130,000 when the agent made each decision on the next 4 steps, our equipment could not support the vast calculations required by MAT (Multi-Agent Transformer) and PPO. In contrast, although the MA2DBT algorithm did not have a prominent performance in optimal strategic exploration, it decomposed the composite action space, which had better stability for strategic optimization in complex scenarios and effectively alleviated the problem of the dimensional complexity caused by the composite action space. In addition, although the use of the DeLighT module reduced the convergence rate, its final effect was slightly stronger than the naïve transformer.

2) MULTI-AGENTS WITH COMPOSITE ACTION SPACE

The single-agent scenario did not fully reflect the advantages of the multi-agent algorithm, so we conducted comparative experiments in a more complex multi-agent environment with composite actions. Using the toy multi-agents reinforcement-learning problems designed by Jiang and Amato [43], the

$$\begin{aligned}
&\mathrm{Var}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t,\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}}\left[\mathbf{Q}_t^i(b)\right]\\
&\quad=\mathrm{Var}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathbf{Q}_t^i(b)|\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}\right]\\
&\quad=\mathrm{Var}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathbb{E}_{\mathbf{a}_t\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\mathbf{Q}_t^i(b)\right]\right]+\mathbb{E}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathrm{Var}_{\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}}\left[\mathbf{Q}_t^i(b)\right]\right]\\
&\quad=\mathrm{Var}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathbb{E}_{\mathbf{a}_t\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\mathbf{Q}_t^i(b)\right]\right]+\mathbb{E}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathrm{Var}_{\mathbf{a}_t^{-i}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}^{-i}}\left[\mathbb{E}_{\mathrm{a}_t^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\mathbf{Q}_t^i(b)\right]\right]+\mathbb{E}_{\mathbf{a}_t^{-i}\sim\pi_{\boldsymbol{\theta}}^{-i}}\left[\mathbf{Var}_{\mathrm{a}_t^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\mathbf{Q}_t^i(b)\right]\right]\right]\\
&\quad=\underbrace{\mathrm{Var}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathbb{E}_{\mathbf{a}_t\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\mathbf{Q}_t^i(b)\right]\right]}_{\text{Variance from state}}+\underbrace{\mathbb{E}_{\mathrm{s}_t\sim d_{\boldsymbol{\theta}}^t}\left[\mathrm{Var}_{\mathbf{a}_t^{-i}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}^{-i}}\left[\mathbb{E}_{\mathrm{a}_t^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\mathbf{Q}_t^i(b)\right]\right]\right]}_{\text{Variance from other agents' actions}}+\underbrace{\mathbb{E}_{\mathbf{a}_t^{-i}\sim\pi_{\boldsymbol{\theta}}^{-i}}\left[\mathbf{Var}_{\mathrm{a}_t^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\mathbf{Q}_t^i(b)\right]\right]}_{\text{Variance from agent i's action}}
\end{aligned}\tag{20}$$

Algorithm 1 Multi-Agent Multi-Action DeLighT Baseline Transformer(MA2DBT)

```
Input: number of agents $n$,number of actions $b$, episodes $K$,ppo iterations $ep$, steps per episode $T$
Initialize : Actor(DelighT Transformer) $\theta_0$,Critic $\phi_0$, Replay buffer $\mathcal{B}$
for $k = 0 \to K - 1$ do
    for $t = 0 \to T - 1$ do
        Collect a sequence of observations $o_t^{i_1}, \ldots, o_t^{i_n}$ from environments.
        Generate representation sequence $\hat{o}_t^{i_1}, \ldots, \hat{o}_t^{i_n}$ by feeding observations to the encoder
        Input $\hat{o}_t^{i_1}, \ldots, \hat{o}_t^{i_n}$ to the decoder.
        for $m = 0 \to n - 1$ do
            for $c = 0 \to b - 1$ do
                Input $a_t^{i_0,j_0}, \ldots, a_t^{i_0,j_c}, \ldots, o_t^{i_m,j_c}$ and infer $a_t^{i_m,j_{c+1}}$ with the decoder.
            end for
        end for
        Execute joint complex actions $a_t^{i_0,j_0}, \ldots, a_t^{i_0,j_b}, \ldots, o_t^{i_n,j_b}$ in environments and collect the reward $R(\mathbf{o}_t, \mathbf{a}_t)$.
        Insert $(\mathbf{o}_t, \mathbf{a}_t, \mathbf{R})$ in to $\mathcal{B}$.
    end for
    for $p = 0 \to ep$ do
        Sample a random batch steps from $\mathcal{B}$.
        Generate $Q_\phi(o^{i_1,a^{j_0}}), .., Q_\phi(o^{i_1,a^{j_b}}), \ldots, Q_\phi(o^{i_n}, a^{j,b})$ with the output layer of the critic.
        Calculate $L_{Critic}(\phi)$ with Equation 22.
        Update the critic by minimising $L_{Critic}(\phi)$ with gradient descent.
        Input $o^{i_1}, .., o^{i_n}$,$s^{i_1}, \ldots, s^{i_n}$ and $a^{-(i_0,j_0)}, \ldots, a^{-(i_0,j_b)}, \ldots, o^{(i_n,-j_b)}$,generate $Q_\phi(o^{i_1,a^{j_0}}), .., Q_\phi(o^{i_1,a^{j_b}}), \ldots, Q_\phi(o^{i_n}, a^{j,b})$ with critic.
        Compute the joint advantage function $\hat{\mathbf{A}}$ based on $Q_\phi(o^{i_1,a^{j_0}}), .., Q_\phi(o^{i_1,a^{j_b}}), \ldots, Q_\phi(o^{i_n}, a^{j,b})$.
        Input $\hat{o}_t^{i_1}, \ldots, \hat{o}_t^{i_n}$ and $a^{i_0,j_0}, \ldots, a^{i_0,j_b}, \ldots, a^{i_n,j_{b-1}}$, generate $\pi_\theta^{i_1,j_1}, \ldots, \pi_\theta^{i_n,j_b}$in at once with the decoder.
        Calculate $L_{Actor}(\theta)$ with Equation 14.
        Update the actor by minimising $L_{Actor}(\theta)$ with gardient descent.
    end for
end for
```

action space was also complex, and the agent needed to decide future multi-step actions one time.

MADDPG was a distributed algorithm. It was equipped with an AC network for each agent, and each network was trained and updated separately. MA2DBT, MAT, and MAPPO were all shared network algorithms. All agents shared a set of network parameters. In addition, as compared to general multi-agent algorithms, the MAT algorithm used the multi-agent decision-making process as a sequential decision-making process, which strengthened the correlations among the decisions of each agent. Here, we compared the MA2DBT algorithm to several multi-agent reinforcement-learning algorithms, such as MAT, MAPPO, and MADDPG, in multi-agent collaborative scenarios and continuously increased the complexity of the composite action space to verify the adaptability of the various algorithms.

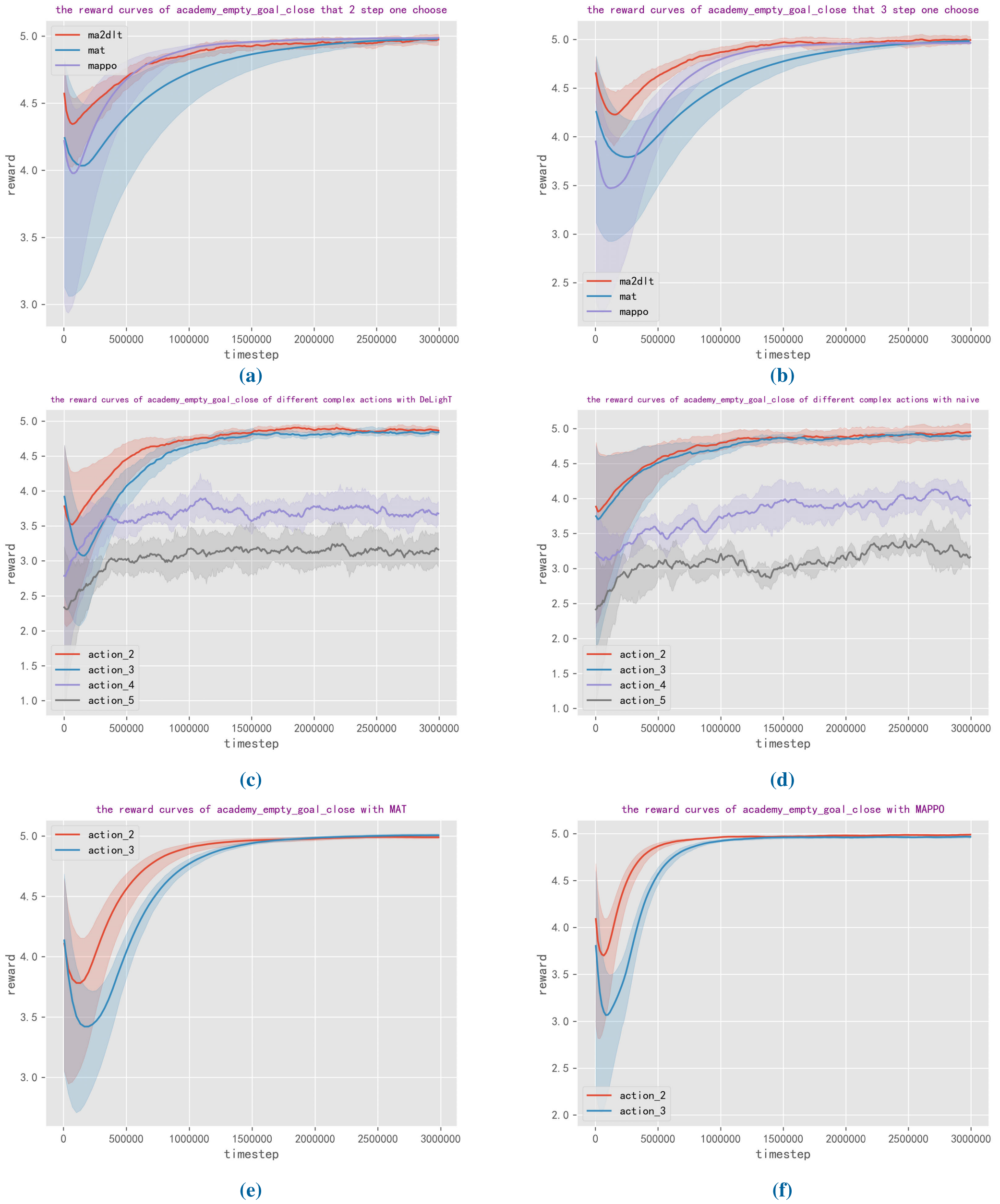
As shown in Figure 5, in the fire-fighter scene, with the gradual expansion of the compound action space, the convergence results of all the algorithms were degraded by varying degrees, but the performance degradation of the MA2DBT algorithm model was the slowest, and the adaptability to the compound action space was the best. With the increased number of steps in a single decision, the gaps between the other algorithms gradually expanded. In the go-together scenario, the MA2DBT algorithm still maintained its optimal performance. In addition, the MAT algorithm achieved better convergence results than the single-decision two-step scenario. This showed that in some scenarios, an agent's singular decision comprising multiple actions was beneficial to the completion of the task, to some extent.

B. ABLATION EXPERIMENT: THE INFLUENCE OF THE PARTS

Next, we explored the strategy of the UAV-swarm escort mission and carried out ablation experiments to determine the contributions of each part of the algorithm to the overall strategic exploration.

1) THE INFLUENCE OF THE DECISION ORDER

For the UAV-swarm escort task, we first explored the influence of the decision-making sequence on the decision-making results. Here, we proposed two multi-agent composite actions as decision-making orders. First, we determined all the actions of an agent according to the sequence of action types and then determined all the actions of the next agent, in turn, until all the decisions were known, which was called the agent priority. The second step was to determine the type


the reward curves of academy_empty_goal_close that 2 step one choose

(a)

the reward curves of academy_empty_goal_close that 3 step one choose

(b)

the reward curves of academy_empty_goal_close of different complex actions with DeLighT

(c)

the reward curves of academy_empty_goal_close of different complex actions with naive

(d)

the reward curves of academy_empty_goal_close with MAT

(e)

the reward curves of academy_empty_goal_close with MAPPO

(f)

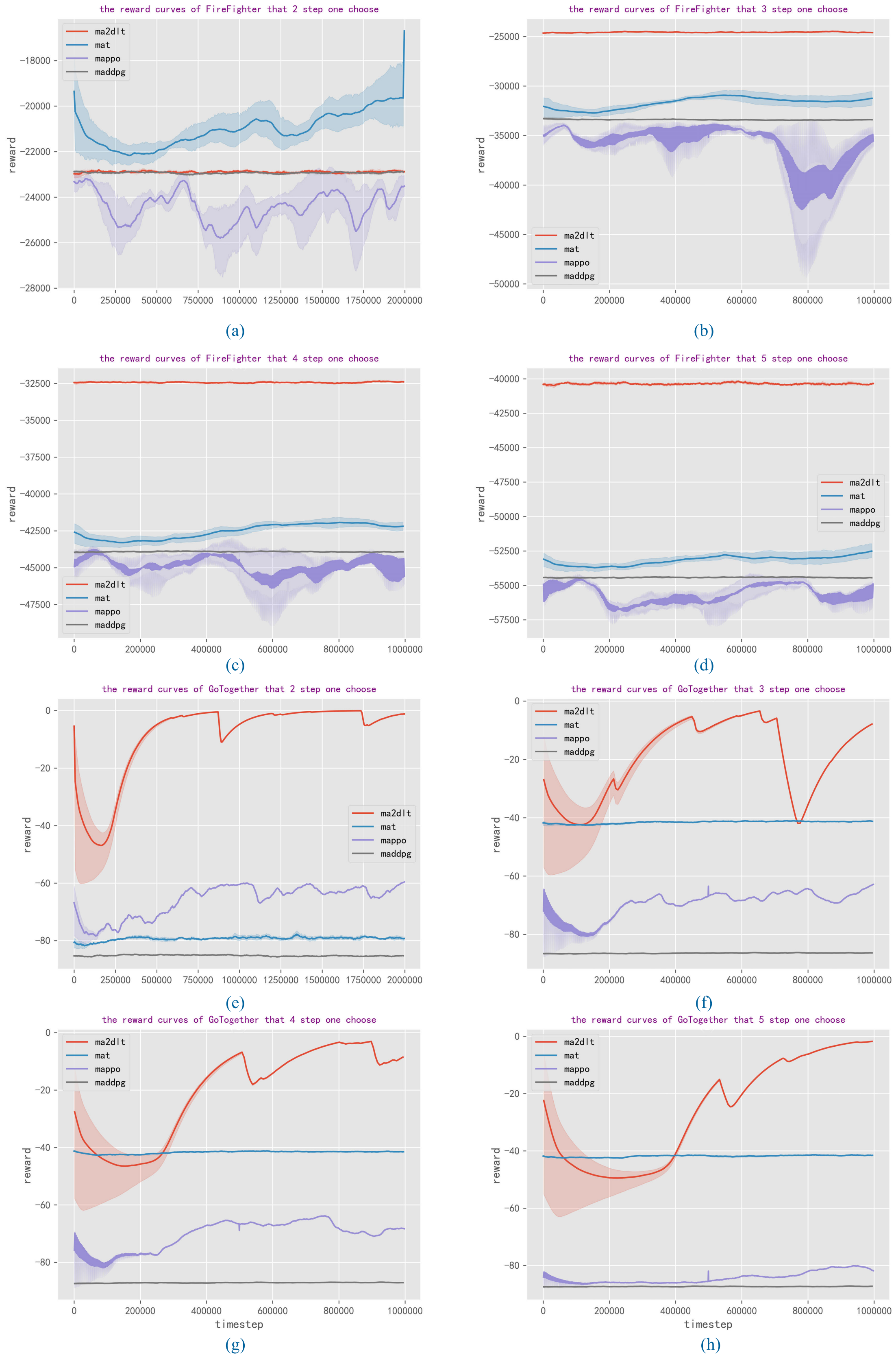
FIGURE 4. **Performance comparison in academic empty-goal scenario. The (a) and (b) are the training results for two-step and three-step decisions, respectively. Images (c)–(f) are the results of DeLighT, a naïve transformer, MAT, and PPO algorithms in this scenario with different steps for each single decision.**

of action decided by all the agents according to the sequence of the agents in order to determine the next type of action of all the agents, in turn, until all the decisions were known, which was called the action priority.

We compared the different decision sequences of the MA2DBT algorithm (the algorithm in the experiment used a naïve version of MA2DBT),as shown Figure 6. The results showed that in our UAV cooperative combat scenario, it was easy to obtain better convergence results when the agent priority approach had been adopted. At the same time, the addition of the DeLighT model enhanced the strategic expression ability of the network model and aided the convergence of the model for better strategic parameters.

2) THE INFLUENCE OF DIFFERENT BASELINES

In our algorithm, the convergence of the strategy depended on the value function obtained by the critic, and the value function calculated by the Monte Carlo method had no deviation but high variance. Therefore, the use of baseline techniques was particularly important. In this phase of the study, we conducted experiments concerning the UAV-swarm



the reward curves of FireFighter that 2 step one choose

(a)

the reward curves of FireFighter that 3 step one choose

(b)

the reward curves of FireFighter that 4 step one choose

(c)

the reward curves of FireFighter that 5 step one choose

(d)

the reward curves of GoTogether that 2 step one choose

(e)

the reward curves of GoTogether that 3 step one choose

(f)

the reward curves of GoTogether that 4 step one choose

(g)

the reward curves of GoTogether that 5 step one choose

(h)

FIGURE 5. Performance comparisons of multi agent scenarios. Images (a)–(d) are the results in a fire-fighter scenario with 2–5 steps determined one time. Images (e)–(h) are the results in a go-together scenario with 2–5 steps determined one time.

TABLE 1. Performance evaluations of sample mission data.

algorithm	singular decision	average episode count/rate					
		located blue	reds scout	scout efficiency[1]	blues guide fire	jamming efficiency[2]	access time[3]
ma2dbt	0.087	5.8(82%)	209	5.6%	0.28	13%	2.8(28%)
ma2dt	0.088	2.03(29%)	77	5.2%	0.11	8%	1.2(12%)
ma2t	0.064	0.2(2.9%)	30	1.2%	0.3	1%	0.1(1%)
mat	0.015	2.48(35%)	106	4.6%	0.26	11%	1.9(19%)
mappo	0.005	0.5(7.1%)	500	0.2%	0.18	0%	0(0%)
maddpg	0.014	0(0%)	76	0%	0.20	0%	0(0%)
vs mat(%)[4]		47%		1%		2%	9%

[1] Scout efficiency is the proportion of the number of reconnaissance results used in cross-positioning in the total number of reconnaissance results.

[2] Jamming efficiency is the proportion of the radar detection capability attenuation caused by UAV interference.

[3] Access time is the maximum maintenance time of the security zone.The task completion requirement is 10s, and the content in parentheses is the task completion degree.

[4] "vs mat" is the increase of ma2dbt relative to mat expressed as percentage.

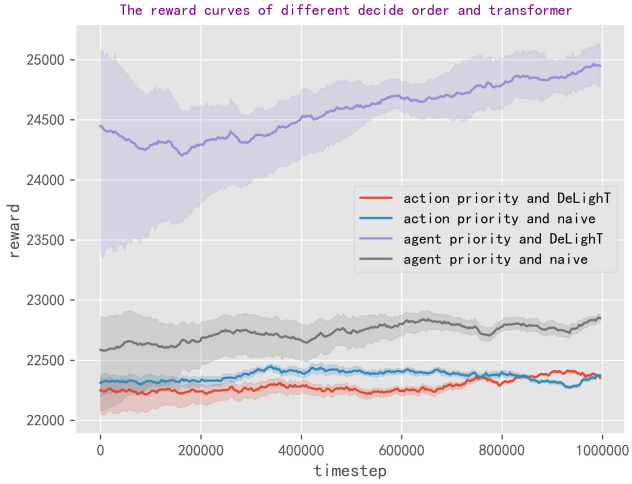


FIGURE 6. Performance comparisons of different sequence in the transformer model.

escort mission to determine the baseline that would best fit the task. In the experiment, X in the loss function of transformer model in the naïve version of MA2DBT algorithm used the advantage value obtained by GAE (Generalized Advantage Estimator) advantage estimation algorithm.

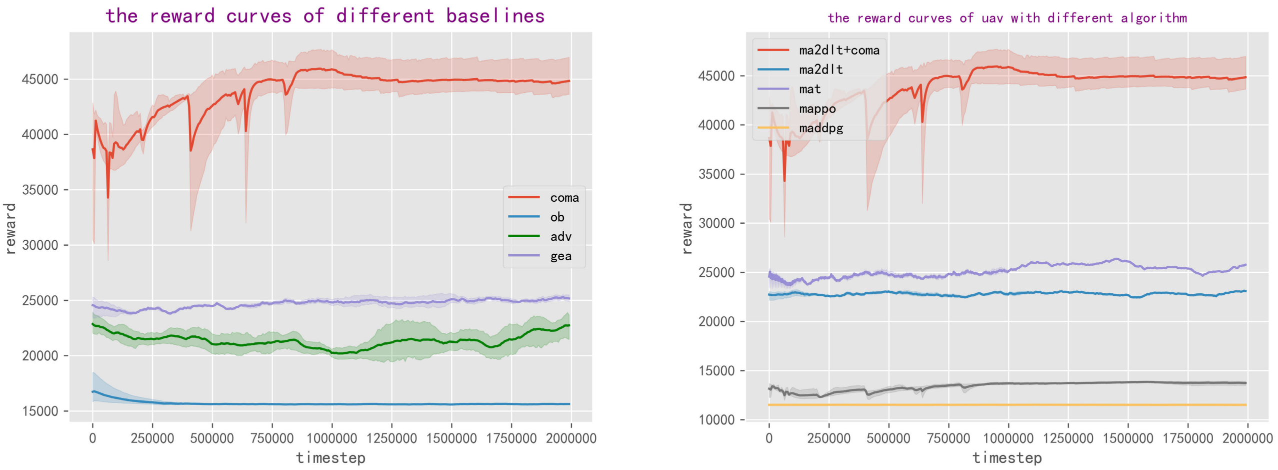
Figure 7 shows that the counterfactual baseline achieved better results in the UAV-swarm cooperative combat scenario. Based on the support of the counterfactual baseline, the MA2DLBT algorithm had better performance than the other algorithms.

The table 1 shows the average data obtained from the same number of evaluations of the decision models for each algorithm, including the single-step decision time and task-related data of the red and blue parties, for each task. We observed that the MA2DBT algorithm had to compromise its efficiency for the single-decision task, but its results were still below 0.1 seconds. As compared to the MAT algorithm, the completion efficiency of the cooperative reconnaissance task increased by 47 %, the detected blue information was nearly doubled, the reconnaissance efficiency increased by 1%, the interference efficiency increased by 2%, and the task completion efficiency increased by 9%. Although the MAPPO algorithm obtained the most reconnaissance information, the coordination between the information was very poor, so that the cross-location could not be completed.

For all aspects of the MA2DBT algorithm, the use of the DeLighT structure further extended the single-step decision time, while the use of the baselines did not have a significant impact. In addition, the DeLighT structure and the addition of the baselines improved the task data obtained by the strategic modelling of the UAV-swarm cooperative combat mission, to some extent.

The requirements of UAV-swarm cooperative escort mission was to reach a safe area and hold the space for more than 10 s. As our results showed, none of the strategy models were able to complete the task. We observed that in this task, flight, reconnaissance, and interference were three dependent sub-tasks, each affecting the others. For example, in general, the number of radar covering the target position was from about three to four. Although the MA2DBT algorithm completed the reconnaissance sub-task well, some reconnaissance and positioning results were not helpful for the subsequent arrival to the safe area. Therefore, the MAT algorithm achieved similar interference efficiency to the MA2DLBT algorithm, with fewer reconnaissance results. We observed that the high degree of the completion of a sub-task did not directly correlate to the completion of the main task. This involved the coordination of objectives in multi-objective tasks. The reasonable allocation of the reward for multiple objectives, the selection of a more appropriate baseline, and the efficiency and feasibility of multi-agents in multi-objective main tasks will be the focus of our next stage of research.

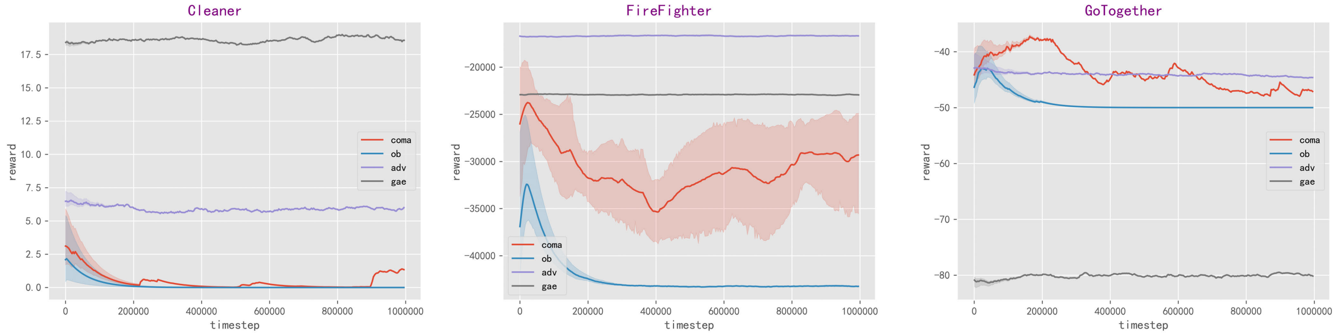
We also did baseline ablation experiments in another training task scenario. The Figure 8 showed that in cleaner and fire-fighter scenarios, the general dominance function



(a)

(b)

FIGURE 7. (a) the performance comparisons of different baselines in MA2DBT. (b) the performance comparisons of different algorithms.



(a)

(b)

(c)

FIGURE 8. Performance comparisons of different baselines. (a) Cleaner, (b) fire fighter, (c) go-together.

had better results, as cleaner tended to estimate the dominance function by the GAE algorithm, while the fire fighter tended to calculate the dominance function according to the definition; in addition, the go-together scenario with a counterfactual baseline achieved better policy convergence results. In summary, we speculated that the choice of the baseline depended on the adaptability of the scene, and specific analyses of specific problems would have better results.

V. CONCLUSION

In this study, we proposed the MA2DBT algorithm to optimize the decision problem of multi-agent compound actions as a long sequential-decision problem while avoiding dimensional complexity. A deeper DeLighT transformer model was used to process the multi-agent sequential data and perform as the actor in decision-making. A critic network was added to estimate the value of all the next actions in a certain agent state. A COMA network was adopted. According to the estimated results, the corresponding baseline was calculated to reduce the estimated variance of the value function and help the actor network update more smoothly.

In addition, we conducted a series of comparative experiments. The experimental results showed that the MA2DBT algorithm had better performance in scenarios in a compound action space, and the advantages of the algorithm were more obvious in multi-agent collaboration scenarios. In the UAV-swarm cooperative escort task, we explored the contribution of each part of the MA2DBT algorithm. The results showed that the priority decision sequence of the agent promoted the convergence of the algorithm for a better strategy. The DeLighT transformer had better strategic exploration and expression ability than the naïve transformer, and the use of counterfactual baselines in the UAV-swarm cooperative task led to the MA2DBT algorithm finding a better strategic model. However, in other scenarios, the general advantage function may have better results, and specific problems would need to be analyzed to determine the use of a baseline. In addition, some experiments showed that in specific scene tasks, compounding the agent actions may have been more conducive to the completion of the tasks.

In the future, we will further optimize the design of the multi-UAV cooperative electronic countermeasures scenario in terms of action space, observation space, reward function, such as designing the agent action space in the form of multi-dimensional discrete space, adding the timing information to the observation vector, etc., and propose corresponding new algorithms in the face of new problems caused by the new design to further improve the level of intelligent decision-making.

APPENDIX
SYMBOL TABLE AND ABBREVIATION TABLE

TABLE 2. Symbol table.

A	Advantage Function
o	Observation
a	Action
π	Policy Function
L	Loss Function
Q	ActionValue Function
V	State Value Function

[1] The main common symbols are listed in this table, and the rest are detailed in the comments of the equation.

TABLE 3. Abbreviation table.

UAV	Unmanned Aerial Vehicle
OODA	Observe-Orient-Decide-Act
AC	Actor-Critic
COMA	Counterfactual Multi-Agent (policy gradients)
RNN	Recurrent Neural Network
CTDE	Centralized Training with Decentralized Execution
MAT	Multi-Agent Transformer
MAPPO	Multi-Agent Proximal Policy Optimization
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MA2DBT	Multi-Agent Multi-Action DeLighT Baseline Transformer
GAE	Generalized Advantage Estimator

REFERENCES

[1] S. Wang, Y. Bao, and Y. Li, "The architecture and technology of cognitive electronic warfare," *Scientia Sinica Informationis*, vol. 48, no. 12, pp. 1603–1613, Dec. 2018.

[2] S. Liu, Z. Lei, Z. Wen, and Y. Ge, "A development review on cognitive electronic warfare," *J. Detect. Control*, vol. 42, no. 5, pp. 1–15, 2020.

[3] O. M. Gul, M. Kulhandjian, B. Kantarci, A. Touazi, C. Ellement, and C. D'amours, "Secure industrial IoT systems via RF fingerprinting under impaired channels with interference and noise," *IEEE Access*, vol. 11, pp. 26289–26307, 2023.

[4] Z. Wang, H. Zhang, H. Zhao, T. J. Cui, and L. Li, "Intelligent electromagnetic metasurface camera: System design and experimental results," *Nanophotonics*, vol. 11, no. 9, pp. 2011–2024, May 2022.

[5] Y. Shuang, L. Li, Z. Wang, M. Wei, and L. Li, "Metasurface-assisted intelligent electromagnetic sensing: Theory, design and experiment," *Chin. J. Radio Sci.*, vol. 36, no. 6, pp. 858–866, 2021.

[6] S. Jia and F. Yang, "Research on intelligent detection method of weak sensing signal based on artificial intelligence," in *Proc. Int. Conf. Adv. Hybrid Inf. Process.*, in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 302, 2019, pp. 90–98.

[7] W. Chi, H. Wang, W. Xie, P. Zhang, and L. Ru, "Research on distributed cooperative intelligent spectrum sensing of UAV cluster," *Wireless Commun. Mob. Com.*, vol. 2022, Jul. 2022, Art. no. 1981533.

[8] S. Rong and L. Jiang, "Application of intelligent optimization methods in jamming resource allocation: A review," *Electron. Opt. Control*, vol. 26, no. 10, pp. 54–61, 2019.

[9] H.-X. Xing, H. Wu, Y. Chen, and K. Wang, "A cooperative interference resource allocation method based on improved firefly algorithm," *Defence Technol.*, vol. 17, no. 4, pp. 1352–1360, Aug. 2021.

[10] J. Tang, X. Chen, X. Zhu, and F. Zhu, "Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1139–1155, Apr. 2023.

[11] M. D. Conway, D. D. Russel, A. Morris, and C. Parry, "Multifunction phased array radar advanced technology demonstrator nearfield test results," in *Proc. IEEE Radar Conf. (RadarConf)*, Apr. 2018, pp. 1412–1415.

[12] F. Ye, X. Li, T. Jiang, Y. Li, and Y. Li, "Research on jamming decision making based on feedback iterative-brown algorithm," in *Proc. IEEE USNC-CNC-URSI North Amer. Radio Sci. Meeting (Joint AP-S Symposium)*, Jul. 2020, pp. 3–4.

[13] B. Zhang and W. Zhu, "Research on decision-making system of cognitive jamming against multifunctional radar," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Sep. 2019, pp. 1–6.

[14] Y. Ran, Y. Cheng, D. Chen, and X. Wang, "Intelligent anti-jamming decision engine based on BP neural network," *J. Signal Process.*, vol. 35, no. 8, pp. 1350–1357, 2019.

[15] F. Ye, Z. Zhou, H. Tian, Q. Sun, Y. Li, and T. Jiang, "Intelligent anti-jamming decision method based on the mutation search artificial bee colony algorithm for wireless systems," in *Proc. USNC-URSI Radio Sci. Meeting (Joint AP-S Symposium)*, Jul. 2019, pp. 27–28.

[16] W. Yu, Y. Sun, X. Wang, K. Li, and J. Luo, "Modeling and analyzing of fire-control radar anti-jamming performance in the complex electromagnetic circumstances," in *Proc. Int. Conf. Man-Mach.-Environ. Syst. Eng.*, in Lecture Notes in Electrical Engineering, vol. 456, 2018, pp. 611–619.

[17] M. Hu, L. Gao, and Z. Zhang, "Comprehensive evaluation on ISAR anti-jamming effectiveness via ELECTRE-III," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Aug. 2013, pp. 979–984.

[18] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[19] Z. Ma, M. He, Z. Liu, L. Gu, and J. Liu, "Survey of unmanned aerial vehicle cooperative control," *J. Comput. Appl.*, vol. 41, no. 5, pp. 1477–1483, 2021.

[20] H. Duan, D. Zhang, Y. Fan, and Y. Deng, "From wolf pack intelligence to UAV swarm cooperative decision-making," *Scientia Sinica Informationis*, vol. 49, no. 1, pp. 112–118, Jan. 2019.

[21] Y. Shen and C. Wei, "Multi-UAV flocking control with individual properties inspired by bird behavior," *Aerosp. Sci. Technol.*, vol. 130, Nov. 2022, Art. no. 107882.

[22] Y. Gao and D. Li, "Unmanned aerial vehicle swarm distributed cooperation method based on situation awareness consensus and its information processing mechanism," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105034.

[23] Y. Zhang, J. Li, B. Hu, and J. Zhang, "An improved PSO algorithm for solving multi-UAV cooperative reconnaissance task decision-making problem," in *Proc. AUS IEEE/CSAA Int. Conf. Aircr. Util. Syst.*, Oct. 2016, pp. 434–437.

[24] L. Yue, R. Yang, Y. Zhang, and J. Zuo, "Research on reinforcement learning-based safe decision-making methodology for multiple unmanned aerial vehicles," *Frontiers Neurorobotics*, vol. 16, Jan. 2023, Art. no. 1105480.

[25] S. Baek and G. York, "Optimal sensor management for multiple target tracking using cooperative unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 1294–1300.

[26] N.-T.-T. Le, "Multi-agent reinforcement learning for traffic congestion on one-way multi-lane highways," *J. Inf. Telecommun.*, vol. 7, no. 3, pp. 255–269, Jul. 2023.

[27] Y. Yifei and S. Lakshminarayanan, "Multi-agent reinforcement learning system for multiloop control of chemical processes," in *Proc. IEEE Int. Symp. Adv. Control Ind. Processes (AdCONIP)*, Aug. 2022, pp. 48–53.

[28] J. Liu, Y. Gu, Y. Cheng, and X. Wang, "Prediction of breast cancer pathogenic genes based on multi-agent reinforcement learning," *Acta Automatica Sinica*, vol. 48, no. 5, pp. 1246–1258, 2022.

[29] S. Do, J. Baek, S. Jun, and C. Lee, "Battlefield environment design for multi-agent reinforcement learning," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2022, pp. 318–319.

[30] I. H. Ahmed, C. Brewitt, I. Carlucho, F. Christianos, M. Dunion, E. Fosong, S. Garcin, S. Guo, B. Gyevnar, T. McInroe, G. Papoudakis, A. Rahman, L. Schafer, M. Tamborski, G. Vecchio, C. Wang, and S. Albrecht, "Deep reinforcement learning for multi-agent interaction," *AI Commun.*, vol. 35, no. 4, pp. 357–368, 2022.

[31] X. Chen, X. Liu, C. Luo, and J. Yin, "Robust multi-agent reinforcement learning for noisy environments," *Peer-to-Peer Netw. Appl.*, vol. 15, no. 2, pp. 1045–1056, Mar. 2022.

[32] J. Riley, R. Calinescu, C. Paterson, D. Kudenko, and A. Banks, "Assured deep multi-agent reinforcement learning for safe robotic systems," in *Proc. Int. Conf. Agents Artif. Intell.*, in Lecture Notes in Computer Science, vol. 13251, 2022, pp. 158–180.

[33] A. Malysheva, D. Kudenko, and A. Shpilman, "MAGNet: Multi-agent graph network for deep multi-agent reinforcement learning," in *Proc. XVI Int. Symp. 'Problems Redundancy Inf. Control Syst.' (REDUNDANCY)*, Oct. 2019, pp. 171–176.

[34] X. Liu and Y. Tan, "Feudal latent space exploration for coordinated multi-agent reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 15, 2022, doi: 10.1109/TNNLS.2022.3146201.

[35] H. Kim, S. Kim, D. Lee, and I. Jang, "Avoiding collaborative paradox in multi-agent reinforcement learning," *ETRI J.*, vol. 43, no. 6, pp. 1004–1012, Dec. 2021.

[36] J. G. Kuba, M. Wen, L. Meng, H. Zhang, D. Mguni, J. Wang, and Y. Yang, "Settling the variance of multi-agent policy gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 13458–13470.

[37] Y. Chen, G. Song, Z. Ye, and X. Jiang, "Scalable and transferable reinforcement learning for multi-agent mixed cooperative–competitive environments based on hierarchical graph attention," *Entropy*, vol. 24, no. 4, p. 563, Apr. 2022.

[38] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15084–15097.

[39] Q. Zheng, A. Zhang, and A. Grover, "Online decision transformer," in *Proc. Mach. Learn. Res.*, Baltimore, MD, USA, vol. 162, 2022, pp. 27042–27059.

[40] L. Meng, M. Wen, C. Le, X. Li, D. Xing, W. Zhang, Y. Wen, H. Zhang, J. Wang, Y. Yang, and B. Xu, "Offline pre-trained multi-agent decision transformer," *Mach. Intell. Res.*, vol. 20, no. 2, pp. 233–248, Apr. 2023.

[41] M. Wen, J. Kuba, R. Lin, W. Zhang, Y. Wen, J. Wang, and Y. Yang, "Multi-agent reinforcement learning is a sequence modeling problem," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 16509–16521.

[42] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 1–9.

[43] S. Jiang and C. Amato, "Multi-agent reinforcement learning with directed exploration and selective memory reuse," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 777–784.

YUAN GAO received the B.S. degree in mechatronics engineering from the North University of China (NUC), in 2019. He is currently pursuing the Ph.D. degree in mechatronics engineering with the Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP), Chinese Academy of Sciences, Changchun, China. His current research interests include intelligent decision making and multi-agents deep reinforcement learning.

YE WANG is currently pursuing the Ph.D. degree with the Chinese Academy of Sciences, Changchun, China. She is currently an Assistant Researcher with the Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP). Her current research interests include intelligent decision making and multi-agents deep reinforcement learning.

LEI ZHANG received the B.S. degree from the University of Jinan, in 2020. He is currently pursuing the M.S. degree in mechatronics engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include reinforcement learning and computer application technology.

LIHONG GUO received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 1999 and 2003, respectively. She is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. Her current research interest includes intelligent computing and its applications.

JIANG LI received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2014. He is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include intelligent simulation and electro-optical system design.

SHOUHONG SUN received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2008 and 2015, respectively. He is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include decision-making AI and intelligent computing and applications.