2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Deep Ad-hoc Sub-Team Partition Learning for Multi-Agent Air Combat Cooperation

Songyuan Fan[1], Haiyin Piao[2], Yi Hu[1], Feng Jiang[1], Roushu Yang[3]

Abstract— **In the future, unmanned autonomous air combat will encounter large-scale confrontation scenarios, where agents must consider complex time-varying relationships among aircraft when making decisions. Previous works have already introduced Multi-Agent Reinforcement Learning (MARL) into air combat and succeeded in surpassing the human expert level. However, they mainly focus on small-scale air combat with low relationship complexity, e.g., 1-vs-1 or 2-vs-2. As more agents join the confrontation, existing algorithms tend to suffer significant performance degradation due to the increase in problem dimensions. In view of this, this paper proposes Deep Ad-hoc Sub-Team Partition Learning(DASPL) to address large-scale air combat problems. DASPL models multi-agent air combat as a graph to handle the complex relations and introduces an automatic partitioning mechanism to generate dynamic sub-teams, which converts the existing large-scale multi-agent air combat cooperation problem into multiple small-scale equivalence problems. Additionally, DASPL incorporates an efficient message passing method among the participating sub-teams.**

I. Introduction

Recently, AI-based air combat has garnered increasing attention from researchers. Numerous methodologies have been proposed to address this complex problem. The existing methods can be summarized into four categories: expert rule[1] , game theory[2] , supervised learning[3] and reinforcement learning[4]. It is worth noting that most of the above approaches have been attempted in small-scale air combat scenarios, such as 1-vs-1 or 2-vs-2, while the problem of air combat in large-scale scenarios is still in the exploratory stage.

Unlike the problems of small-scale air combat that have been solved, large-scale air combat is often accompanied by complex relationships among the participants. These relationships often show the characteristics of large numbers, multiple relationships, and high order. If the relationship among participants is taken into account without simplification, the adoption of fully connected communication will bring a lot of message redundancy and the performance will be significantly degraded[5]. It is worth mentioning that large-scale air combat often shows a strong local correlation. Aircraft with high correlation can often achieve unity of objectives within a certain period, which indicates that we can divide the agents into several sub-teams based on their relevance. This has the potential to significantly enhance the collaborative efficiency in large-scale air combat scenarios.

In view of this, we propose a method named **D**eep **A**d-hoc **S**ub-Team **P**artition **L**earning(DASPL) for large-scale air combat, which performs well in multiple scenarios. Concretely, DASPL abstracts air combat into a graph structure and utilizes Sub-Team Partition Learning (SPL) to divide it into multiple sub-teams at intervals. Furthermore, DASPL designs a message-passing mechanism to ensure effective communication between sub-teams by increasing the lower bound of mutual information variation. In this way, it can extremely promote intra-group coordination and facilitate inter-group cooperation. Finally, DASPL uses the deep hybrid action network to achieve action decisions. Overall, the contributions of DASPL are summarized as follows:

1. A deep ad-hoc sub-team partition learning mechanism is innovatively proposed for enhancing the ability to adapt to unfixed large-quantity air combat for the first time. This approach can dynamically divide the formation into multiple sub-teams, which plays a crucial role in dealing with complex air combat.

2. Based on the sub-teams, we design a variational message summarizer for efficient and economical message passing.

3. Experiments show that DASPL has achieved good results in comparison with other SOTA Multi-Agent Reinforcement Learning algorithms. In large-scale scenarios such as 6-vs-6 or 8-vs-8, the reward is at least about 28.3% higher than other algorithms.

II. Related Works

A. Air Combat AI

Air combat is a representative complex multi-agent game problem that has been studied since the 1970s. Early research mainly uses expert knowledge to solve air combat problems [6]. Recent research pays attention to the application of Reinforcement Learning(RL). Some researchers try to apply MADRL algorithms like Qmix[7], MAPPO[8], and DyMA-CL[9] to air-combat since these algorithms have demonstrated good effects in various games schemes, such as SC2, and Dota. Large-scale air combat has a high similarity with game environments mentioned above. Ding and Yang [10] took advantage function of air combat as a reward function and designed a dynamic fuzzy Q-learning model for maneuver decision in air combat. Zhang [11] achieved

[1]S. Fan, Y. Hu and F. Jiang are with the Department of Computing, Harbin Institute of Technology, Harbin, China. email:SongYuan-Fan@outlook.com; email:huyi_hit@163.com; email:haiqizhu@hit.edu.cn; email:fjiang@hit.edu.cn

[2]H. Piao is with the School of Electronics and Information, Northwestern Polytechnical University, Xian, China. e-mail: haiyinpiao@mail.nwpu.edu.cn

[3]R. Shu is with SAIL, Shenyang, China. e-mail: Roushu_Yang@outlook.com

the effect of air combat strategy that combines heuristic exploration and random exploration of air combat strategy. To solve the multi-agent decision-making problem, Piao [12] built a hierarchical policy network to deal with complex discrete/continuous maneuvers. However, the methods mentioned above mainly focus on small-scale air combat. As the scale of the problem increases, the cooperation relationship becomes very complex. It is difficult to apply the existing methods directly to large-scale air combat scenarios.

B. Sub-Team Partition Learning in MADRL

Many efforts have been made for automatic grouping in multi-agent systems. One class of related works predefine specific responsibilities for each agent based on goals, visibility, or other factors that require domain knowledge or priori settings. Another class of approaches focuses on individual roles by training deep models. Among them, SOG [13] provides zero-shot generalization ability to the dynamic number of agents and the varying partial observability by electing conductors and setting message summarizers. ROMA [14] learns dynamic roles that depend on the context agents observe. RODE [15] decomposes the joint action spaces and integrates the action effects into the role policies to boost learning. GoMARL [16] learns automatic grouping without domain knowledge, factorizing the joint action values as a combination of group-wise values. In our work, DAPSL inherits the advantages of these two main approaches and proposes a grouping method that uses domain knowledge to train models.

III. Proposed Method

A. Overview

In large-scale air combat scenarios, it is necessary to determine which opponents are the primary targets, and which teammates can collaborate with each other. Consequently, as illustrated in Fig. 1, we abstract all the above key procedures into three important modules named Sub-Team Partition Learning(SPL), Bi-level Message Passing(BMP), and Cooperative Tactics Decision-Making(CTD). More details about the three modules are provided as follows.

B. Sub-Team Partition Learning(SPL)

In real-world air combat, commanders usually divide it into several formations. Inspired by this, DASPL proposes a grouping method named Sub-Team Partition Learning (SPL), which achieves automatic grouping by training a model combining Inductive Logic Programming(ILP) and representation learning. Currently, most multi-agent grouping methods utilize prior or domain knowledge for simple rule writing. Compared with hard-coded rules, learning methods have more advantages in scalability and flexibility. Therefore, DASPL employs a combination of ILP and representation learning to fully exploit the advantages of both. As previously mentioned, each node in the air combat graph corresponds to an aircraft entity. Naturally, we first need to describe the node attributes in the graph. The node vector for aircraft i at time t is represented as o_t^i. Based on the characteristics of air combat, we have defined 11 predicates for ILP. In order to summarize the rules of which conditions should be or not be grouped together, we utilize Hierarchical Transformer Networks(HTN) for rules generation and the correctness is verified by label datas.

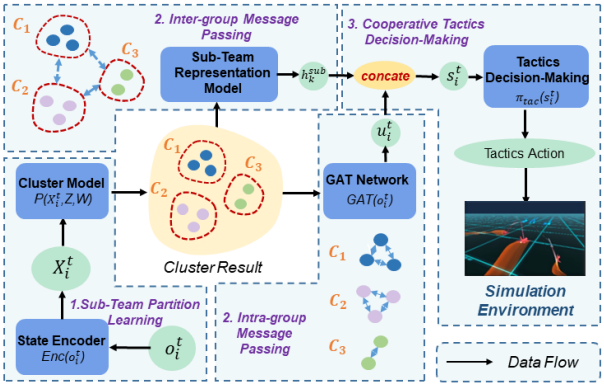


Fig. 1. The network architecture of DASPL is composed of three modules: (1) Sub-Team Partition Learning(SPL) divides some sub-teams and generates a series of graphs; (2) Bi-level Message Passing(BMP) reaches a tactical agreement by communicating the latent combat intention among inter-group and intra-group; (3) Cooperative Tactics Decision-Making(CTD) makes tactic decisions.

Based on the rules above, we construct a dataset containing several triples of $S = [x^+, x, x^-]$, where x^+ and x represent two agents that need to be closely coordinated and should be placed in the same sub-team, whereas x^- is an agent that should be placed in different sub-team with x^+ and x. From the perspective of feature engineering, we attempt to encode the agents belonging to the same group like x^+ and x through a neural network. After being projected a high-dimensional space, the distance between them is closer and the distance from x^- is farther. dis_- is defined as the distance between sample x or x^+ and x^-, dis_+ is defined as the distance between sample x and x^+. Enc is a multi-layer fully connected network responsible for mapping sample features to high-dimensional space. By mapping the agent state o_i^t to a high-dimensional space $X_i^t = Enc(o_i^t)$, the model Enc has the ability to represent corresponding semantics in a high-dimensional space. This model uses $Triplet\ Loss$ for training, and the loss function is defined as $loss_{mi}$.

M agents constitute a set $X = [X_1^t, X_2^t, \cdots, X_M^t]$, where $X_i^t = [x_{i,1}, x_{i,2}, \cdots, x_{i,n}]$ indicating that each agent is described by an n-dimensional attribute, and $Z = [Z_1^t, Z_2^t, \cdots, Z_k^t]$ is a set of k cluster centers, where $Z_i^t = [z_{i,1}, z_{i,2}, \cdots, z_{i,n}]$. Then k-means attempts to divide M agents into k sub-teams by minimizing the formula P.

$$P(X, Z) = \sum_{l=1}^{k} \sum_{i=1}^{M} \sum_{j=1}^{n} u_{i,l} d(x_{i,j}, z_{l,j})$$
$$d(x_{i,j}, z_{l,j}) = (x_{i,j} - z_{l,j})^2 \quad (1)$$

The so-called noise dimension appears, which makes the

clustering algorithm take the distance of the noise dimension into account when calculating the distance between samples, resulting in a decrease in clustering accuracy. Therefore, we use a weighted cluster algorithm to overcome the impact of noise dimensions on cluster accuracy. Initialize a weight value for each feature dimension, and when the objective function converges, the weight corresponding to the noise dimension will tend to zero, making it possible to ignore the impact of the noise dimension when calculating the distance between samples. A weight parameter w_j^β is added to the original k-means constraint function to adjust the impact of each dimension on the clustering results through different weight values. β is a hyperparameter, the optimization goal for weighted clustering is shown in Equation (2).

$$\hat{P}(X,Z) = \sum_{l=1}^{k}\sum_{i=1}^{M}\sum_{j=1}^{n} u_{i,l} w_j^\beta d(x_{i,j}, z_{l,j})$$
$$d(x_{i,j}, z_{l,j}) = (x_{i,j} - z_{l,j})^2 \quad (2)$$

As we can see, D_j is actually the sum of the distances of all sample points on the $j-$th dimension until convergence. In this way, agents with similar features will be divided into the same sub-team, while agents with farther features will be divided into different sub-teams. We can obtain a set of k sub-teams $C = [C_1, C_2, \cdots, C_k]$.

C. Bi-Level Message Passing(BMP)

The above work led to several sub-teams through SPL. As a representative of complex multi-agent systems, it is crucial to improve multi-agent strategies by handling the communication message properly. To enhance the global collaboration of agents, we propose a bi-level message passing mechanism based on formed sub-teams.

Intra-group Node Level Message Passing: For each node in a sub-team, other nodes' information can be obtained by using the graph aggregation model. Graph Attention Networks(GAT) is currently an important way to process graph data information transmission and has achieved good results in many benchmarks. Inspired by recent efforts which use variational inference as a regular term to estimate the state in hidden variable spaces, we propose a message passing method enable the agents summarize their local observations into brief latent variables, which can extract valuable information from the whole trajectory.

Specifically, Let $\tau_t^i = (o_{t+1}^i, a_{t+1}^i, \cdots, o_{t+T-1}^i, a_{t+T-1}^i)$ indicates the agent i's trajectory of future T-steps. The information conveyed is a random Gaussian variable of C dimensions sampled from an encoder, i.e., $h_t^i = f_\varphi(o_t^i)$. We maximize the mutual information between h_t^i and τ_t^i conditioned on o_t^i. Increasing mutual information between the trajectory τ_t^i and observation o_t^i by continuously raising the lower bound of variation is shown in Equation (3). Finally, we utilize Graph Attention Networks to achieve intra-group message aggregation $\mu_t^i = GAT(h_t^i, ..., h_t^j)$.

$$I(h_t^i; \tau_t^i | o_t^i) \geq E_{h_t^i, \tau_t^i, o_t^i}\big[\log\big(q_\phi(h_t^i|o_t^i, \tau_t^i)\big)\big] + H(h_t^i|o_t^i) \quad (3)$$

where $q_\phi(.)$ is the variational estimator and defines the opposite of the lower bound as L_{NL}.

Inter-group Sub-Team Level Message Passing: Considering that air combat is modeled as a partially observable environment, node level messaging introduces valuable information by other agents, but the information brought within the sub-team is far from sufficient. Message sharing should also be carried out among sub-teams to form a unified will among formations and cooperate to achieve established goals.

Since the entire graph contains rich situational information, we can use it to enhance collaboration and coordination capabilities. Therefore, we attempt to make the sub-team features extracted by the sub-team feature encoder $f_\theta(.)$ contain more situational information from other sub-teams. We also introduce variational inference to inter-group message passing, increasing mutual information between the entire graph situation and sub-team representation by continuously raising the lower bound of the variational inference.

Analogous to node level, o^{Graph} indicates features of the entire graph. h_K^{Sub} is the representation of the k-th sub-team, i.e., $h_k^{Sub} = f_\theta(o_t^i, \cdots, o_j^t)$. We try to increase mutual information between the entire graph o^{Graph} and the k-th sub-team h_K^{Sub} by continuously raising the lower bound of the variational inference conditioned on $o_t^i, \cdots, o_t^j$.

$$I(h_K^{Sub}; o^{Graph} | o_t^i ... o_t^j) \geq E_{h_K^{Sub}, o^{Graph}, o_t^i ... o_t^j}$$
$$\big[\log\big(q_\delta(h_K^{Sub}|o_t^i ... o_t^j, o^{Graph})\big)\big] + H(h_K^{Sub}|o_t^i ... o_t^j) \quad (4)$$

$q_\delta(.)$ is the variational estimator and defines the opposite of the lower bound as L_{TL}.

Intuitive understanding, the entire graph contains all available situation information. Increasing mutual information between the sub-team representation and the entire graph representation can effectively enable each sub-team to infer situation information from other sub-teams or even the entire graph and combine its own representation to form a global situation awareness.

D. Cooperative Tactics Decision-Making(CTD)

Collaborative Tactical Decisions(CTD) π_{tac} will perform corresponding action output according to the fusion information s_t^i. CTD outputs the aircraft's actions from three aspects, namely, shooting decisions, maneuver actions, and target decisions. We have preset 14 discrete maneuver actions commonly used in air combat. Target decisions command the relative maneuvering targets of each aircraft. All three aspects are discrete action spaces. The rewards for the simulation environment include event and round type. Event rewards are mainly designed for key events in air combat such as missile launches, aircraft stall control, anti boundary crash, radar/missile locking, and unlocking. Round rewards represent rewards that can only be obtained at the end of air combat, mainly covering global events such as shooting down enemy aircraft, being shot down, team victory/failure, and so on.

Specifically, DASPL maintains two separate neural networks: a policy network(referred to as an actor) with parameters θ and a value function network (referred to as a critic) with parameters ϕ. These networks are shared among all agents. The critic network is a 5-layer perception, denoted as $V(s_t^i;\phi)$ mapping $s_t^i \rightarrow R$,where $s_t^i = h_k^{Sub}||\mu_t^i||o_t^i$. The advantage $\hat{A}$ is computed through the Generalized Advantage Estimation (GAE) method. For the actor-network π_{tac}, π_{tac} utilizes Multi-Agent Proximal Policy Optimization (MAPPO) for optimization training. The main optimization objective is to maximize:

$$L_{RL}(\theta) = -\frac{1}{Bn}\sum_{i=1}^{B}\sum_{t=1}^{n}\min\left(r_t^i(\theta)\hat{A}_t^i, clip(r_t^i(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t^i\right) \tag{5}$$

where $r_t^i(\theta) = \frac{\pi_{tac}(a_t^i|s_t^i;\theta)}{\pi_{tac}(a_t^i|s_t^i;\theta^{old})}$ and θ^{old} is the parameters of CTD before the update, θ is the parameters of CTD and $\hat{A}$ is the advantage function. Therefore, the optimization function is shown in Equation (5). The overall loss can be written as:

$$L_{All} = L_{RL} + \beta_1 L_{TL} + \beta_2 L_{NL} \tag{6}$$

where β_1 and β_2 are hyper-parameters.

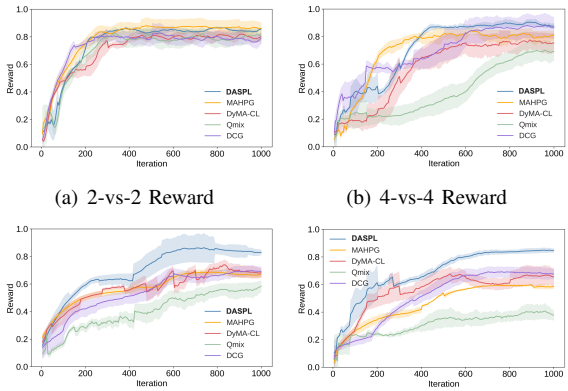
IV. Experiments

This chapter commences with an introduction to the experimental environment and scenario settings. Subsequently, comprehensive and convincing experimental results are presented to address the following research questions: (1) RQ1: Can DASPL significantly enhance learning performance compared to the state-of-the-art MADRL algorithms? (2) RQ2: Does the performance of DASPL noticeably deteriorate when certain key components are removed from the model? (3) RQ3: Can DASPL accurately and dynamically partition corresponding sub-teams according to predefined relationships at critical moments? (4) RQ4: Can DASPL generate policies that resemble those developed by human experts?

A. Experiment Setup

Simulation Environment: The experiments were conducted within WUKONG[17], an air combat simulation environment based on real aircraft dynamics, which offers extensive configurable items, such as aircraft quantity, aircraft model, missile situation, initial battle conditions, scenarios involving beyond visual range(BVR) or dogfight air combat and so on. This sophisticated simulation environment serves as a robust platform for validating the efficacy of the algorithms employed, ensuring a fair and scientifically sound assessment of their performance. To validate the efficacy of our method within the air combat simulation environment, this study employs red and blue sides as air combat scenarios. The aircraft models, early warning information, and missiles of both sides on both sides are identical.

Baselines: To reveal the learning performance and the ability of our method, we conducted a comparative analysis against four state-of-the-art Multi-Agent Deep Reinforcement Learning (MADRL) techniques. The baseline methods



(a) 2-vs-2 Reward (b) 4-vs-4 Reward

(c) 6-vs-6 Reward (d) 8-vs-8 Reward

Fig. 2. Learning curves with baseline algorithms: Each curve represents the normalized reward of the corresponding method

are listed as follows: (1) MAHPG[17] is a well-adopted baseline in air combat, which is capable of learning various strategies and transcending expert cognition by adversarial self-play learning. (2) Qmix[7] is a novel value-based method that can train decentralized policies in a centralized end-vs-end fashion. (3) DyMA-CL [9]starts learning from small-scale multi-agent scenarios by course and solves large-scale multi-agent problems by gradually increasing the number of agents. (4) DCG[18] is a well-adopted baseline for graph-based value function decomposition. The joint action value is decomposed through a coordination graph that only includes pairwise correlation, and the joint decision is implemented in combination with belief propagation, thereby solving the relative overgeneralization problem in multi-agent scenarios.

B. Learning Performance Comparison with Baseline Algorithms(RQ1)

To evaluate whether DASPL can yield substantial enhancements in learning performance compared to state-of-the-art MADRL algorithms and to demonstrate its effectiveness, we conducted a comparative analysis against the aforementioned four baseline methods. During this experiment, all baselines underwent training with 20,000 self-play samples for each iteration within the WUKONG environment.

Specifically, the experiment encompasses four scenarios: 2-vs-2, 4-vs-4, 6-vs-6, and 8-vs-8. Figure 2 displays the normalized combat rewards of various methods (the shaded region represents the standard deviation of the average evaluations over five trials).

Investigating Figure 3 (a), under the 2-vs-2 scenarios, due to the small size of the problem and the relatively simple nature of the problem, the five algorithms can converge from the initial 0.1 to about 0.8. DASPL does not exhibit strong advantages compared to other methods. As the number of agents increases, five algorithms are affected to varying degrees in the 4-vs-4 scenarios. In particular, Qmix suffers a substantial decline in terms of convergence speed and final score, as illustrated in Figure 3 (b). Some algorithm performance degradation is more evident in the 6-

TABLE I

DISPLAY OF WIN RATES IN DIFFERENT SCENARIOS

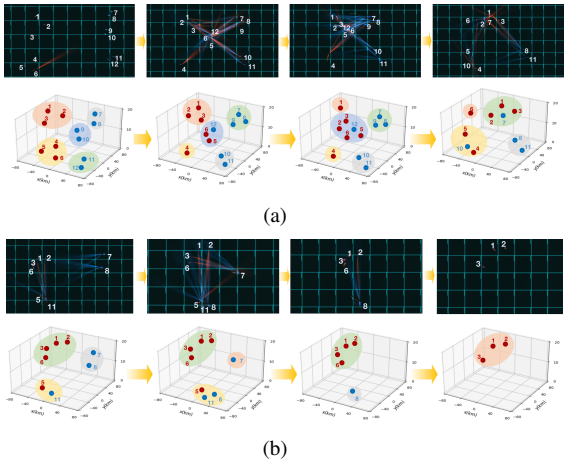
Two Message Passing Layers							
Clusters	1	2	4	6	8	12	16
2V2	0.70±0.09	**0.83±0.12**	0.65±0.11				
4V4	0.62±0.11	0.68±0.06	**0.70±0.09**	0.62±0.07	0.58±0.09		
6V6	0.61±0.09	0.64±0.08	0.72±0.08	**0.84±0.11**	0.73±0.07	0.57±0.11	
8V8	0.60±0.06	0.62±0.10	0.67±0.06	**0.81±0.13**	0.75±0.12	0.62±0.09	0.58±0.08
average	0.63±0.09	0.69±0.09	0.69±0.09	**0.76±0.10**	0.69±0.09	0.60±0.10	0.58±0.08
One Message Passing Layers							
Clusters	1	2	4	6	8	12	16
2V2	0.64±0.12	**0.79±0.09**	0.60±0.08				
4V4	0.58±0.09	0.67±0.06	**0.68±0.07**	0.62±0.08	0.55±0.09		
6V6	0.59±0.08	0.59±0.12	0.68±0.11	**0.81±0.09**	0.63±0.11	0.52±0.09	
8V8	0.57±0.07	0.58±0.08	0.66±0.11	**0.78±0.11**	0.74±0.08	0.60±0.07	0.57±0.09
average	0.59±0.09	0.66±0.06	0.66±0.09	**0.73±0.09**	0.64±0.09	0.56±0.08	0.57±0.09
Zero Message Passing Layers							
Clusters	1	2	4	6	8	12	16
2V2	0.52±0.08	**0.69±0.07**	0.52±0.06				
4V4	0.56±0.09	**0.64±0.08**	0.64±0.06	0.55±0.11	0.48±0.12		
6V6	0.54±0.08	0.55±0.07	0.67±0.11	**0.74±0.12**	0.56±0.09	0.47±0.12	
8V8	0.54±0.08	0.53±0.09	0.62±0.06	0.68±0.11	**0.70±0.10**	0.53±0.07	0.47±0.06
average	0.54±0.08	0.60±0.07	0.62±0.10	**0.65±0.11**	0.58±0.10	0.50±0.10	0.47±0.06

vs-6 and 8-vs-8 scenarios. DyMA-CL, MAHPG, and DCG exhibit a final convergence score of approximately 0.6. In contrast, DASPL demonstrates outstanding performance on large-scale problems, achieving remarkable results in both convergence speed and final score which is at least about 28.3% higher and 32.9% higher than other algorithms on the scale of 6-vs-6 and 8-vs-8 problems.

To sum up, DASPL demonstrates superior adaptability across various scales of air combat problems, efficiently acquiring and mastering corresponding air combat skills. Particularly noteworthy is its extraordinary ability in solving large-scale air combat scenarios (6-vs-6 and 8-vs-8), showcasing a more robust policy compared to other algorithms. While MAHPG, DyMA-CL, and DCG show gradual increases in cumulative returns, their learning speed and final states notably lag behind DASPL. An analysis of DyMA-CL reveals its limitation in addressing large-scale air combat scenarios due to its inherent complexity. Course learning equips DyMA-CL with fundamental combat skills suitable for smaller-scale environments, yet these skills struggle to adapt to high-dimensional, dynamic scenarios, resulting in relatively generalized performance.

C. Ablation of Deep Ad-hoc Sub-Team Partition Learning (RQ2)

We conducted an ablation study to ascertain the efficacy of Ad-hoc Sub-Team Partition Learning and Bi-Level Message Passing within the DASPL framework. Several components were deliberately removed to evaluate the impact of these mechanisms on solving air combat problems. This experiment aims to answer whether the performance of DASPL experiences a noticeable decline when removing certain key components from the model.



(a)

(b)

Fig. 3. Case Study

As previously mentioned, Deep Ad-hoc Sub Team Partition Learning can be divided into several sub-teams based on predefined situations in complex air combat scenarios. To evaluate the impact of Deep Ad-hoc Sub Team Partition Learning on the effectiveness of large-scale air combat algorithms, we conducted ablation experiments. Following the principle of controlling variables, we designed three experimental groups: one without message passing (Message Passing Layers set to 0), another with Node-Level message passing (Message Passing Layers set to 1), and the last with Bi-Level message passing (Message Passing Layers set to 2), and assessed their performance individually. To ensure equitable and robust testing, we employed a bot named 'very hard' within the WUKONG simulation environment, programmed based on insights from expert pilots' experiences. Utilizing identical training samples and reward mechanisms, our trained models faced this 'very hard' bot in 1000 confrontations, guaranteeing credible outcome assessments. The win rate calculation method involves dividing the number of wins by the total games played.

To ensure Ad-hoc Sub-Team Partition Learning works. We can fix the number of Message Passing Layers and the scale of the scenario. Take 2-vs-2 scenarios with 2 Message Passing Layers scheme as an example, when k-means $k = 1$, it indicates that no sub-teams will be conducted, and the win rate against the bot is 70%. When $k = 2$, the win rate against the bot increases to 83%. When $k = 4$, that is, each aircraft forms its sub-team, the win rate against the bot decreases to 65%. This trend persists in the other scenarios, where increasing k initially elevates the win rate but eventually leads to a decline, as depicted in Table I.

To confirm whether Bi-Level Message Passing is effective, we can fix the cluster of k-means and the scale of the scenario. Take 2-vs-2 scenarios with k-means $k = 1$ scheme as an example, when Message Passing Layers is set to 2, which means take Bi-Level Message Passing, the win rate against the bot is 70%. When Message Passing Layers is set to 1, the win rate decreases to 64%. When Message

Passing Layers is set to 0, the win rate decreases to 52%. This trend persists in the other scenarios, where Bi-Level Message Passing gets the best result, as depicted in Table I.

D. Correctness Analysis of Ad-hoc Sub-Team Partition Learning(RQ3 and RQ4)

To answer RQ3 and RQ4, verify that DASPL can accurately identify predefined situations in dynamic and complex air combat environments, and subsequently assign them to the same sub-team. Our analysis delves into 8 pivotal instances occurring within both 6-vs-6 scenarios.

At the outset of an air combat scenario, DASPL initially partitions 12 agents into 5 sub-teams following a grouping structure of 2 or 3 aircraft per sub-team, with 1 # (# represents the aircraft number), 2 # and 3 # as a sub-team, 4 #, 5 # and 6 # as a sub-team. This initial grouping primarily relies on proximity, grouping agents with relatively closer distances to maintain depth in the attack during the initial phase of confrontation.

Subsequently, aircraft 5 #, 6 # coordinated attacks 12#, so DASPL adjusted and grouped 5 #, 6 #, and 12 # into the same sub-team. As air combat progressed, aircraft 5 #, 6 #, 2 #, and 3 # executed an encirclement tactic around 12 #, initiating missile launches that successfully eliminated both aircraft 12 # and 9 #. This strategic move by aircraft 5 #, 6 #, 2 #, and 3 # on the red side showcased their cohesive approach, exhibiting robust cooperative capabilities. Then, they gradually formed two encirclement patterns: one involving 4 #, 5 #, and 10 #, and another encompassing 1 #, 2 #, 3 #, and 7 #, as shown in Figure 3(a).

From the perspective of the red team, illustrated in Figure 3(b), aircraft 5 # on the red side adopts an extremely advantageous tail attack posture against the blue side 11 #, and then the blue side 8 # promptly supports 11 #. DASPL intelligently groups the blue side 8 #, 11 #, and red side 5 # into the same sub-team based on avoidance tactics. At this time, the red side strategically focuses solely on engaging the blue side 8 # and 11 #, and the red side 5 # seizes the opportune moment and successfully eliminates both aircraft 8 # and 11 #.

The red side ultimately secures victory by sacrificing 4 #, 5 #, and 6 # to win. This particular case study vividly demonstrates DASPL's ability to adeptly recognize and analyze prevailing situations, subsequently effectuating sub-team formations based on predefined rules. In general, due to the complex sub-team division of air combat is particularly similar to human strategic thinking, it is reasonable to infer that DASPL can understand complex air combat relationships and emerging expert-level air combat methods such as encirclement tactics and collaborative attack tactics.

V. Conclusion

In this paper, aimed at solving multi-agent cooperative air combat tasks in large-scale scenarios, we propose a practical algorithm called Deep Ad-hoc Sub-Team Partition Learning. It can adeptly address the questions in air combat from the top intention to the bottom action control by partitioning groups and constructing effective communication between intra-sub-team and inter-sub-team. The favorable outcomes we got may guide how to use Multi-Agent Deep Reinforcement Learning to conduct large-scale air combat. In this work, we used prior knowledge to partition groups so we will try to eliminate this reliance and make our algorithm more intelligent in our further research.

References

[1] M. J. Kim and I. Han, "The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms," *Expert Syst. Appl.*, vol. 25, no. 4, pp. 637–646, 2003.

[2] J. K. D. Morrison, R. J. Hansman, and S. Sgouridis, "Game Theory Analysis of the Impact of Single-Aisle Aircraft Competition on Emissions," *J. Aircr.*, vol. 49, no. 2, p. p.483-494, 2012.

[3] X. U. Ximeng, R. Yang, and F. U. Ying, "Situation assessment for air combat based on novel semi-supervised naive Bayes," *J. Syst. Eng. Electron.*, vol. 29, no. 4, pp. 768–779, 2018.

[4] Q. Yang, J. Zhang, G. Shi, J. Hu, and Y. Wu, "Maneuver Decision of UAV in Short-Range Air Combat Based on Deep Reinforcement Learning," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2019.

[5] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-agent game abstraction via graph attention neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 7211–7218.

[6] K. GOODRICH and J. MCMANUS, "Development of a tactical guidance research and evaluation system (TGRES)," in *Flight simulation technologies conference and exhibit*, 1989, p. 3312.

[7] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 7234–7284, 2020.

[8] C. Yu *et al.*, "The surprising effectiveness of ppo in cooperative multi-agent games," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24611–24624, 2022.

[9] W. Wang *et al.*, "From few to more: Large-scale dynamic multiagent curriculum learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 7293–7300.

[10] L. Ding and Q. Yang, "Research on air combat maneuver decision of UAVs based on reinforcement learning," *Avion. Technol.*, vol. 49, no. 2, pp. 29–35, 2018.

[11] X. Zhang, G. Liu, C. Yang, and J. Wu, "Research on air confrontation maneuver decision-making method based on reinforcement learning," *Electronics*, vol. 7, no. 11, p. 279, 2018.

[12] Z. Sun *et al.*, "Multi-agent hierarchical policy gradient for air combat tactics emergence via self-play," *Eng. Appl. Artif. Intell.*, vol. 98, p. 104112, 2021.

[13] J. Shao, Z. Lou, H. Zhang, Y. Jiang, S. He, and X. Ji, "Self-Organized Group for Cooperative Multi-agent Reinforcement Learning," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 5711–5723, 2022.

[14] T. Wang, H. Dong, V. Lesser, and C. Zhang, "Roma: Multi-agent reinforcement learning with emergent roles," *ArXiv Prepr. ArXiv200308039*, 2020.

[15] T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang, "Rode: Learning roles to decompose multi-agent tasks," *ArXiv Prepr. ArXiv201001523*, 2020.

[16] Y. Zang *et al.*, "Automatic grouping for efficient cooperative multi-agent reinforcement learning," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[17] Z. Sun, H. Piao, Z. Yang, Y. Zhao, G. Zhan, D. Zhou, G. Meng, H. Chen,X. Chen, B. Qu, and Y. Lu, "Multi-agent hierarchical policy gradient for Air Combat Tactics emergence via self-play," Eng. Appl. Artif. Intell.,vol. 98, pp. 104–112, Feb. 2021.

[18] W. B¨ohmer, V. Kurin, and S. Whiteson, "Deep coordination graphs," in 37th Int. Conf. Mach. Learn. (ICML 2020), vol. 119, Elect. Netw., Jul.2020.