Shapley-Enhanced Mean Field Multi-Agent Reinforcement Learning for Multi-UAV Pursuit-Evasion Maneuvering Decision-Making

Yunxiao Guo[1,2,†], Dan Xu[2,†], Han Long[1,†], Shaofei Chen[1], Chaoyang Chen[3], Chang Wang[1,*]

***Abstract*— With the development of artificial intelligence, Multi-Agent Deep Reinforcement Learning (MADRL)- based UAV autonomous maneuver decision methods have made significant progress, enabling swarm systems to respond effectively to complex pursuit-evasion tasks. However, the current MADRL-based maneuver decision methods still face three main challenges: the curse of dimensionality, credit assignment, and a non-stationary environment. To address these issues, this paper proposes the Shapley Mean Field Multi-Agent Reinforcement Learning (SMFMARL) algorithm for UAV swarm pursuit-evasion maneuver decisions. Specifically, we use Approximate Marginal Contribution (AMC) to estimate the Shapley-Q value for evaluating the real contribution of each UAV in the swarm. We also introduce a mean-field representation into the SQV to mitigate the non-stationarity caused by the AMC estimator and reduce the dimension of the input action. To improve the sample efficiency, this paper designs a training framework that introduces the self-play and experience replay mechanism, which can enhance the UAV's reuse of high-quality samples. Furthermore, we present a comprehensive pursuit-evasion reward that comprises catch, defensive, and advantage functions, guiding the UAV in deciding whether to pursuit or evade in different situations. In numerical simulations, we compare the proposed method with state-of-the-art MADRL algorithms in a 15-UAV swarm air-pursuit evasion scenario. The results show that SMFMARL can emerge with complex strategies.**

I. Introduction

With their low cost, flexible maneuverability, and high operational efficiency, unmanned aerial vehicle (UAV) swarms are playing a vital role in low-altitude economy applications [1]. In urban airspace management, ensuring UAV security has emerged as a critical research challenge. This security issue can be formulated as a pursuit-evasion problem between UAV swarms. Within such an autonomous pursuit-evasion system, maneuver decision-making represents the most critical component, directly determining the overall performance of the UAVs [2].

Deep reinforcement learning (DRL) has been applied in various fields to realize high-performance end-to-end learning for specific tasks [3]–[5]. In recent years, DRL and Multi-Agent DRL (MADRL) were sequentially introduced into the UAV swarm maneuver decision-making to deal with the multi-UAV pursuit evasion tasks, which has greatly improved the UAVs' maneuver performance [3], [4].

However, as the number of UAVs continues to increase, the action and state space dimensions will grow exponentially, leading to performance collapse and increased training time in MADRL [6]. To alleviate it, mean field theory emerged in the MADRL-based maneuver decision algorithms [7]–[10]. By decoding the joint action with mean field representation and replacing the Q value with mean field approximation, Yang et al. [7] proposed the Mean Field Multi-Agent Reinforcement Learning (MFMARL) algorithm, which reduced the dimension of the action space representation in a large-scale multi-agent system. To develop a robust version of MFMARL, Jusup et al. [10] developed a model-based MFMARL algorithm that attains safe policies even in unknown transitions, outperforming the vehicle repositioning problem. Nevertheless, the mean-field representation cannot identify the essential interactions, thereby increasing training time, especially in non-stationary environments [9]. Therefore, Wang et al. [8] proposed the adaptive mean field Q-learning algorithm by deriving the mean-field approximation with attention-based adaptive weight, which can guide the UAV to pay attention to the high-performance UAVs.

Another crucial challenge of MADRL is the credit assignment problem; the contribution of individual agents is hard to evaluate accurately, in multi-agent systems [11]. To handle the credit assignment, Wang et al. [12] first introduced the Shapley value into the MADRL method, and proposed the SQDDPG algorithm that evaluates the Shapley-Q-value (SQV) of the agent to represent their contribution. Based on the Shapley-Q-value, Li et al. [13] considered Shapley counterfactual credit assignment that introduces the counterfactual method to the Shapley value. Han et al. [14] further studied the theoretical problem of Shapley value-based reward reallocation, proving it is efficient and stable in convex games. However, the current methods depend on AMC to estimate the Shapley value; it masks the actions of agents who do not exist in the coalition with zeros [12], which drops some information about other agents and increases the non-stationarity.

Inspired by the mean-field and Shapley value-based MADRL methods, this paper considers the mean-field representation and the Shapley value in MADRL-based pursuit evasion maneuver decision, alleviating the performance collapse while input dimension increases and the credit

[1] Y. Guo, H. Long, S. Chen, and C. Wang are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: guoyunxiao.nudt@hotmail.com; longhan@nudt.edu.cn; chenshaofei01@nudt.edu.cn; wangchang07@nudt.edu.cn).

[2] D. Xu is with the School of System Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: xudan_changsha@163.com).

[3] C. Chen is with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: cychen@hnust.edu.cn).

[†] These authors contributed equally to this work.

[*] Corresponding author: Chang Wang.

This work was supported by the National Natural Science Foundation of China under Grants 62576354 and 62533010.

assignment problem. The main contributions are shown as follows:

- We propose the Shapley Mean Field Multi-Agent Reinforcement Learning (SMFMARL) algorithm as an UAV swarm pursuit-evasion maneuver decision method, by using AMC to estimate the SQV for evaluating the real contribution of each UAV in the swarm, and introducing mean-field representation into the SQV to avoid the non-stationarity caused by the masked agents' actions.
- We design a multi-UAV red-blue confrontation environment, which introduces the self-play and experience replay mechanism into the training framework to reuse the samples produced by both sides. In this framework, we further show a comprehensive pursuit evasion reward that consists of catch, defensive, and advantage reward functions, which can guide UAV to take pursuit or defense correctly in different situations.

II. Background

A. Mean Field Multi-Agent Reinforcement Learning

The Mean Field Multi-Agent Reinforcement Learning (MFMARL) algorithm considers a multi-agent system consisting of M agents with N dimensional discrete action space, for any j-th agent and at time t, the mean field action can be represented as follows:

$$\overline{\boldsymbol{a}}_{j,t} = \frac{1}{|\mathcal{N}_j|} \sum_{i \in \mathcal{N}_j} \boldsymbol{a}_{j,t}^i, \tag{1}$$

where the $\boldsymbol{a}_{j,t}^i$ is the one-hot coded action of i-th agent observed by j-th agent. According to Eq. (1), $\boldsymbol{a}_{j,t}^i$ can be rewritten as the mean field action add a fluctuation $\delta \boldsymbol{a}_{j,t}^i$:

$$\boldsymbol{a}_{j,t}^i = \overline{\boldsymbol{a}}_{j,t} + \delta \boldsymbol{a}_{j,t}^i, \tag{2}$$

The mean field actions are one-hot coded and then calculated by averaging. Therefore, the i-th row of $\overline{\boldsymbol{a}}_{j,t}$ can be interpreted as the empirical distribution of the i-th action taken by $|\mathcal{N}_j|$ neighbors around j-th agent. By using the mean field representation, the Q-value function with pairwise approximation can be approximated as follows [7]:

$$Q^j\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = \frac{1}{|\mathcal{N}_j|} \sum_{i \in \mathcal{N}_j} Q^j\left(\boldsymbol{s}_t, \boldsymbol{a}_{j,t}, \boldsymbol{a}_{j,t}^i\right) \approx Q^j\left(\boldsymbol{s}_t, \boldsymbol{a}_{j,t}, \overline{\boldsymbol{a}}_{j,t}\right). \tag{3}$$

This approximation simplified the action input during the estimation of the Q-value function, which alleviated the performance collapse while the agent number increased. When the Q-value is determined, the MFMARL update the agent's policy $\boldsymbol{\pi}_t^j\left(\boldsymbol{a}_{j,t} \mid s_t, \overline{\boldsymbol{a}}_{j,t}\right)$ based on the following Bolzmann distribution expression:

$$\pi_t^j\left(\boldsymbol{a}_j \mid \boldsymbol{s}, \overline{\boldsymbol{a}}_j\right) = \frac{\exp\left(\beta Q_t^j\left(\boldsymbol{s}, \boldsymbol{a}_j, \overline{\boldsymbol{a}}_j\right)\right)}{\sum_{\boldsymbol{a}_j'} \exp\left(\beta Q_t^j\left(\boldsymbol{s}, \boldsymbol{a}'_j, \overline{\boldsymbol{a}}_j\right)\right)}. \tag{4}$$

Furthermore, the corresponding mean field V-value function expression based on Bellman equation can be derived as below:

$$V^j\left(\boldsymbol{s}_t\right) = \sum_{\boldsymbol{a}_{j,t}} \pi_t^j\left(\boldsymbol{a}_j \mid \boldsymbol{s}, \overline{\boldsymbol{a}}_j\right) \mathbb{E}_{\overline{\boldsymbol{a}}_j\left(\boldsymbol{a}^{\neg j}\right)}\left[Q_t^j\left(\boldsymbol{s}_t, \boldsymbol{a}'_j, \overline{\boldsymbol{a}}_j\right)\right], \tag{5}$$

where $\boldsymbol{a}^{\neg j}$ represents the joint action that does not include the j-th action of agents and $\overline{\boldsymbol{a}}_{\boldsymbol{j}}(\boldsymbol{a}^{\neg j})$ is the corresponding mean field action after removing it. The MFMARL guides the agents in generating transitions from the new policies, calculating the mean field action, and estimating the value function, until convergence.

In conclusion, MFMARL replaces the joint action representation with mean field representation, the action space of the j-th agent can be reduced from $|\mathcal{N}_j| \times M$ dimension to $M \times N$ dimension, alleviating the dimensional curse caused by the increase in the number of agents especially when the observable agents number is significant larger than the action space: $|N_j| \gg N$.

B. Shapley Value Based Multi-Agent Reinforcement Learning

The Shapley value is proposed to quantify each player's contributions by calculating the marginal contribution of the coalition, in a cooperative game. For any agent $i \in \mathcal{N}$, given a coalition $\mathcal{C}$ that does not include the i-th agent, denote the set of agents in the multi-agent system as $\mathcal{N}$, the Shapley value is defined as follows:

$$\mathrm{Sh}_i = \sum_{\mathcal{C} \subseteq \mathcal{N} \backslash \{i\}} \frac{|\mathcal{C}|!(|\mathcal{N}| - |\mathcal{C}| - 1)!}{|\mathcal{N}|!} \delta_i(\mathcal{C}), \tag{6}$$

where $\delta_i(\mathcal{C}) = v(\mathcal{C} \cup i) - v(\mathcal{C})$ represents the marginal contribution of the i-th agent in the coalition $\mathcal{C}$. To tackle the credit assignment in MADRL, Shapley value is introduced as the Shapley Q Value (SQV) function. Wang et al. [12] utilize sampling principle to propose the Approximate Marginal Contribution (AMC) method to estimate the Shapley Q value function of the agent:

$$Q_{\Phi_i}\left(\boldsymbol{s}, \boldsymbol{a}_i\right) \approx \frac{1}{K} \sum_{j=1}^{K} \hat{\Phi}_i\left(\boldsymbol{s}, \boldsymbol{a}_{\mathcal{C}_j \cup (i)}\right), \forall \mathcal{C}_j \sim \Pr(\mathcal{C} \mid \mathcal{N} \backslash \{i\}), \tag{7}$$

where $\widehat{\Phi}_i\left(\boldsymbol{s}, \boldsymbol{a}_{\mathcal{C}, \cup (i)}\right) : \mathcal{S} \times \mathcal{A}_{\mathcal{C} \cup \{i\}} \rightarrow \mathbb{R}$ is a function to approximate the marginal contribution directly. K is the sampling frequency and the coalition $\mathcal{C}$ is formed randomly by model $\Pr(\mathcal{C}|\mathcal{N}\backslash\{i\})$. Subsequently, the SQV-base algorithms continuously estimate and iteratively improve the SQV to update each agent's policy and optimize its decision to enhance its contribution. Meanwhile, introducing SQV improves the multi-agent system's interpretability because the Shapley value is interpretable when measuring marginal contribution.

C. Multi-UAV Swarm Pursuit-Evasion Environment

Consider a fixed-wing UAV swarm red-blue confrontation scenario, denote the red and blue UAV camps as $\mathbf{UAV}^{(r)} = \left\{\mathrm{UAV}_1^{(r)}, \mathrm{UAV}_2^{(r)}, \cdots, \mathrm{UAV}_n^{(r)}\right\}$ and

$\mathbf{UAV}^{(b)} = \left\{\mathrm{UAV}_1^{(b)}, \mathrm{UAV}_2^{(b)}, \cdots, \mathrm{UAV}_n^{(b)}\right\}$, respectively. The $\mathrm{UAV}_j^{(r)}$ represents j-th UAV in the red camp, and $\mathrm{UAV}_i^{(b)}$ represents i-th UAV in the blue camp. With the fixed attitude assumption and random noise compensation, the kinematics model of the fixed-wing UAV is usually simplified to the 4 Degree-of-Freedom (DoF) [15]. The kinematics model of the single UAV can be represented as follows:

$$\dot{\boldsymbol{\xi}}^i = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x^i \\ y^i \\ \psi^i \\ \phi^i \end{bmatrix} = \begin{bmatrix} v^i\cos\psi^i + \eta^{x_i} \\ v^i\sin\psi^i + \eta^{y^i} \\ -\left(\alpha_{\mathrm{g}}/v^i\right)\tan\phi^i + \eta_{\psi^i} \\ f(\phi,\phi_d) \end{bmatrix}, \quad (8)$$

where $(x^i, y^i) \in \mathbb{R}^2$ is planar the position of UAV i, $\psi^i \in \mathbb{S}^1$ is the heading, and determined by the roll angle $\phi^i \in \mathbb{S}^1$; α_g is the gravity acceleration; η is the random noise, which is follow the Gaussian distribution.

Next, to illustrate the observation of the UAV, we take an example with $\mathrm{UAV}_j^{(r)}$ against $\mathrm{UAV}_i^{(b)}$, the coordinate of $\mathrm{UAV}_j^{(r)}$ at time t is $\overrightarrow{\boldsymbol{r}}_{j,t}^{(r)} = (x_{j,t}^{(r)}, y_{j,t}^{(r)})$ and that of $\mathrm{UAV}_i^{(b)}$ is $\overrightarrow{\boldsymbol{r}}_{i,t}^{(b)} = (x_{i,t}^{(b)}, y_{i,t}^{(b)})$, then the position vector $\overrightarrow{\boldsymbol{r}}_{ji,t}^{(r)}$ and Euclidean distance $\Delta r_{ji,t}$ between two UAVs can be represented as $\overrightarrow{\boldsymbol{r}}_{ji,t}^{(r)} = (x_{j,t}^{(r)}, y_{j,t}^{(r)}) - (x_{i,t}^{(b)}, y_{i,t}^{(b)}), \Delta\boldsymbol{r}_{ji,t} = \|\Delta\overrightarrow{\boldsymbol{r}}_{ji,t}^{(r)}\|$, respectively. Similarly, we can define the velocity vector $\overrightarrow{\boldsymbol{v}}_{j,t}^{(r)}$ and its Euclidean distance $\Delta\boldsymbol{v}_{ji}^{(r)}$, respectively. By using the velocity direction $\frac{\overrightarrow{\boldsymbol{v}}_{j,t}^{(r)}}{\|\overrightarrow{\boldsymbol{v}}_{j,t}^{(r)}\|}$ of $\mathrm{UAV}_j^{(r)}$ as the baseline, the angle between two UAVs based on the baseline can be represented as the following form:

$$\varphi_{ji,t}^{(r)} = \arccos\frac{\Delta\overrightarrow{\boldsymbol{r}}_{ji,t}^{(r)} \cdot \overrightarrow{\boldsymbol{v}}_{j,t}^{(r)}}{\Delta\boldsymbol{r}_{ji,t}\left\|\overrightarrow{\boldsymbol{v}}_{j,t}^{(r)}\right\|} \quad (9)$$

Furthermore, we assume each UAV is equipped with a radar, which can observe the information of no more than n_{obs} UAVs within the radius R_{radar}. Denote $D_{catch} < R_{radar}$ is the maximum pursuit distance, and φ_{max} is the maximum angle of pursuit, when the distance Δr and angle φ between two opposite UAVs are within the ranges of D_{catch} and φ_{max}, respectively, the UAV that takes advantage will launch a capture net and probabilistically catch the opposite UAV. The full-observation of i-th UAV can be described as: $\boldsymbol{s}_{j,t}^{(r)} = \begin{bmatrix} \varsigma_{j,t}^{(1)} & \varsigma_{j,t}^{(2)} & \cdots & \varsigma_{j,t}^{(n_{obs})} \end{bmatrix}$, where $\varsigma_{j,t}^{(i)}$ represents the i-th observed UAV. For each $\varsigma_{j.t}$, we select the following six variable that the UAVs are concerned in pursuit evasion: $\varsigma_{j,t}^{(i)} = [\Delta x_{ji}^{(r)}, \Delta y_{ji}^{(r)}, \psi_{ji}^{(r)}, \varphi_{ji}^{(r)}, \Delta\boldsymbol{v}_{ji}^{(r)}, \Delta\boldsymbol{r}_{ji}]^\top$.

Denote the action that UAV execute at time t as $\boldsymbol{a}_t = [a_{\phi,t}, a_{v,t}]$, where $a_{\phi,t} \in [-1,1], a_{v,t} \in [-1,1]$ represent the action in roll angle and velocity, respectively. Then the roll angle and velocity will be updated by the following equations:

$$\phi_{t+1} = \begin{cases} \phi_{\max}, & \text{if } \phi_t + a_\phi\Delta\phi\Delta t \geq \phi_{\max} \\ -\phi_{\max}, & \text{if } \phi_t + a_\phi\Delta\phi\Delta t \leq -\phi_{\max} \\ \phi_t + a_\phi\Delta\phi\Delta t, & \text{otherwise} \end{cases}, \quad (10)$$

$$v_{t+1} = \begin{cases} v_{\max}, & \text{if } v_t + a_v\Delta v\Delta t \geq v_{\max} \\ v_{\min}, & \text{if } v_t + a_v\Delta v\Delta t \leq v_{\min} \\ v_t + a_v\Delta v\Delta t, & \text{otherwise} \end{cases}, \quad (11)$$

where Δt denotes the minimum timestep, $\Delta\phi$ and Δv represent the step of the roll angle and velocity, respectively.

III. Our Approach

A. Shapley Mean Field Multi-Agent Reinforcement Learning

The Shapley value-based methods can evaluate the agents' contribution by estimating their Shapley value in the MARL system [16]. In SQDDPG, the actions from different coalitions will be frequently sampled and concatenated to evaluate the SQV. When the size of coalitions is different, the dimensions of concatenated actions will be different. The current methods, such as Approximate Marginal Contribution (AMC), also fulfill the concatenated actions with zeros to keep the input's dimension consistency [12].

However, this method faces two challenges: The first is that AMC has a low extent of ability; as the agent number increases, AMC should retrain the model from scratch. The second is that AMC can not deal with the curse of dimensionality challenge; it fulfills the actions of agents who are not in the coalition with zeros to keep the fixed input dimension, which wastes the input dimension. Meanwhile, the mean-field approaches can not only deal with the dimensional curse when the number of agents increases but also keep a fixed input dimension along as the action space does not change [6]. In addition, due to the actions with mean field representations also representing the action distribution among the agents, it is natural to extend this as the number of agents increases.

Therefore, we introduce the mean-field action representation into the Shapley Q value and propose the Shapley mean-field Q Value as follows:

$$\begin{aligned} &Q_{\mathrm{MF}}^{\mathrm{sh}}(\boldsymbol{s}, \boldsymbol{a}_i) \\ &= \sum_{\mathcal{C}\subseteq\mathcal{N}_i\setminus\{i\}} \frac{|\mathcal{C}|!\left(|\mathcal{N}_i| - |\mathcal{C}| - 1\right)!}{|\mathcal{N}_i|!} \cdot \Big[Q_t^i\left(\boldsymbol{s}, \overline{\boldsymbol{a}}_i^{\mathcal{C}\cup\{i\}}\right) \\ &\quad - Q_t^i\left(\boldsymbol{s}, \overline{\boldsymbol{a}}_i^{\mathcal{C}}\right)\Big], \end{aligned} \quad (12)$$

where $\overline{\boldsymbol{a}}_i^{\mathcal{C}_k}$ can be calculated by following equation:

$$\overline{\boldsymbol{a}}_i^{\mathcal{C}_k} = \frac{1}{|\mathcal{C}_k|}\sum_{j\in\mathcal{C}_k}\boldsymbol{a}_i^j. \quad (13)$$

By applying AMC on the Shapley mean-field Q value, we can obtain its sampling approximation:

$$\begin{aligned} &Q_{\mathrm{MF}}^{\mathrm{sh}}(\boldsymbol{s}, \boldsymbol{a}_i) \\ &\approx \frac{1}{M}\sum_{k=1}^{M}\left[Q_t^i\left(\boldsymbol{s}, \overline{\boldsymbol{a}}_i^{\mathcal{C}_k\cup\{i\}}\right) - Q_t^i\left(\boldsymbol{s}, \overline{\boldsymbol{a}}_i^{\mathcal{C}_k}\right)\right], \\ &\forall\mathcal{C}_k \sim Pr(\mathcal{C}|\mathcal{N}\setminus\{i\}), \end{aligned} \quad (14)$$

By replacing the joint action with a mean-field action, the dimension inconsistency with action inputs in the Shapley

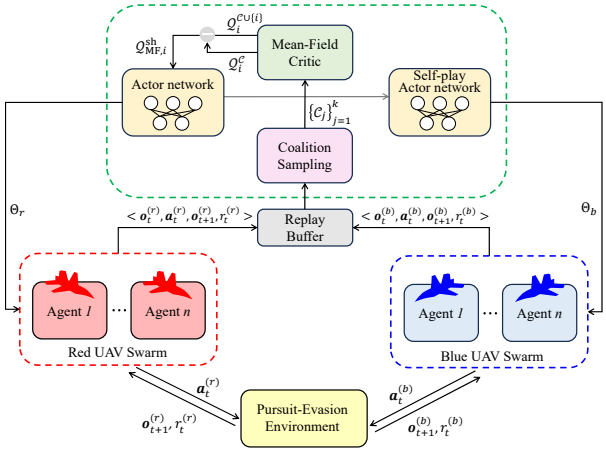

Fig. 1: The schematic diagram of the Shapley Mean-Field Multi-Agent Reinforcement Learning algorithm

Q value is solved because the dimension of the mean-field action is independent of the number of agents. In addition, the dimension of action space observed by the j-th agent can be reduced from $|\mathcal{N}_j| \times M$ to $M \times N$, and then the curse of dimensionality caused by the number of agents in MADRL can be mitigated. Moreover, it has the most significant dimension reduction effect when $|\mathcal{N}_j| \gg N$.

Based on the Shapley mean-field Q value, we propose the Shapley Mean-Field Multi-Agent Reinforcement Learning algorithm for maneuver decision in UAV swarm pursuit evasion (See Fig. 1).

As shown in Fig. 1, in the training stage, the two camps' UAV agents interact with the pursuit evasion environment designed in the last section, observe the environment, and make decisions simultaneously. The swarm pursuit evasion will terminate when one of the camp's UAV swarms is caught, or the preset maximum episode length is reached. To improve the utilization efficiency of the collected samples, the transition data $< \boldsymbol{o}, \boldsymbol{a}, \boldsymbol{o}', r >$ of two camps UAVs will be stored in the same replay buffer after each turn of action execution.

After collecting enough transition data, the experience replay mechanism will sample the transitions from the replay buffer. Meanwhile, the algorithm will generate a set $\{\mathcal{C}_j\}_{j=1}^{j}$ that contains k coalitions stochastically, and the actions of sampled data will be converted to the mean-field action of the corresponding coalition $\mathcal{C}_j$ by using Eq. (13).

Moreover, while the adversarial UAV agents update their policies simultaneously, the previous buffer data will be retired. To alleviate this, we applied the self-play mechanism with a fixed timestep in the proposed algorithm. Specifically, the adversarial agents will be replaced at a fixed timestep instead of being updated along with the training agents. From the curriculum perspective, self-play with fixed-timestep updates can provide a gradual curriculum for the pursuit evasion system, increasing learning efficiency.

B. Reward Scheme

To alleviate the sparse reward problem, we design an incentive

According to the pursuit judgment, a matrix $\boldsymbol{\Gamma}_t^{(r)}$ which indicates the catch state of the red and blue sides at the t moment is designed as follows. In the case of the red UAVs, it is denoted as: $\boldsymbol{\Gamma}_t^{(r)} = \{\Gamma_{ij,t}^{(r)}\}$, where $\Gamma_{ji,t}^{(r)}$ represent the pursuit state of $\mathrm{UAV}_j^{(r)}$ against $\mathrm{UAV}_i^{(b)}$:

$$\Gamma_{ji,t}^{(\mathrm{r})} = \begin{cases} 0, \varphi_{ji,t}^{(r)} \geq \varphi_{\max} \quad \text{or} \quad \Delta r_{ji} \geq D_{catch} \\ 1, \varphi_{ji,t}^{(r)} < \varphi_{\max}, \Delta r_{ji} < D_{\mathrm{catch}}, P\left(\varpi_{ji}\right) < \mu \\ 2, \varphi_{ji,t}^{(r)} < \varphi_{\max}, \Delta r_{ji} < D_{catch}, P\left(\varpi_{ji}\right) \geq \mu \end{cases} \tag{15}$$

If $\Gamma_{ji,t}^{(r)} = 0$, it shows the $\mathrm{UAV}_j^{(r)}$ has not locked $\mathrm{UAV}_i^{(b)}$; If $\Gamma_{ji,t}^{(r)} = 1$, it shows the $\mathrm{UAV}_j^{(r)}$ has locked $\mathrm{UAV}_i^{(b)}$ but has not caught; If $\Gamma_{ji,t}^{(r)} = 2$, it shows the $\mathrm{UAV}_j^{(r)}$ has caught $\mathrm{UAV}_i^{(b)}$.

Taking $\mathrm{UAV}_j^{(r)}$ as an example, the reward function is divided into two parts: catch reward and defensive reward. In the catch reward, the reward for catching an enemy UAV is set to 5, the reward for locking on an enemy UAV is 0.2, and if it only maneuvers, it will receive -0.05 as a penalty. The catch reward can be expressed as:

$$r_{j,t,catch}^{(r)} = \begin{cases} 5, \Gamma_{ji,t}^{(\mathrm{r})} = 2 \\ 0.2, \Gamma_{ji,t}^{(\mathrm{r})} = 1 \\ -0.05, \Gamma_{ji,t}^{(\mathrm{r})} = 0 \end{cases}, \forall i \leq n_{obs} \in \mathbb{Z}^+ \tag{16}$$

Correspondingly, the defensive part of the reward can be designed as 0.05 for not being locked or caught by an enemy UAV, -5 for being caught by an enemy UAV, and -0.2 for being locked by an enemy UAV. The defensive reward can be expressed as:

$$r_{j,t,def}^{(r)} = \begin{cases} -5, \Gamma_{ijj,t}^{(b)} = 2, \exists i \\ -0.2, \Gamma_{ij,t}^{(b)} = 1, \exists i \\ 0.05, \Gamma_{ij,t}^{(b)} = 0, \forall i \end{cases} \tag{17}$$

The above reward functions are sparse, which will decrease the learning efficiency [17]. To improve the learning efficiency of the UAV swarm, we design an incentive reward function named pursuit evasion advantage reward:

$$\begin{aligned} & r_{j,t,adv}^{(r)} \\ =& \frac{1}{n_{obs}} \sum_{i=1}^{n_{obs}} \left[\frac{\Delta \boldsymbol{r}_{ji}}{\sum_{k=1}^{n_{obs}} \Delta \boldsymbol{r}_{jk}} \left(\frac{\varphi_{max} - \varphi_{ji,t}^{(r)}}{2\pi} + \frac{\varphi_{ij,t}^{(b)} - \varphi_{max}}{2\pi} \right) \right. \\ & \left. \cdot \frac{D_{catch} - \Delta \boldsymbol{r}_{ji}}{D_{catch}} \right] \end{aligned} \tag{18}$$

where $\frac{\varphi_{max} - \varphi_{ji,t}^{(r)}}{2\pi}$ indicates whether the i-th enemy UAV is in the j-th UAV's pursuit angle range; the $\frac{\varphi_{ij,t}^{(b)} - \varphi_{max}}{2\pi}$ indicates whether the j-th UAV is in the j-th enemy UAV's pursuit angle range; the $\frac{D_{catch} - \Delta \boldsymbol{r}_{ji}}{D_{catch}}$ indicates the UAVs are within the pursuit range of each other.

The physical interpretation of the pursuit evasion advantage reward function can be concluded as follows: When the enemy UAVs are within the pursuit angle range and out of the enemy UAV's pursuit angle, the advantage reward function will be positive and encourages the UAVs to approach the enemy UAV and adjust the angle to lock the enemy UAV. As the UAV approaches the enemy UAV, the $\Delta \boldsymbol{r}_{ji}$ will decrease, and the advantage reward (18) will increase. When the UAVs are in the range of the enemy's pursuit angle and are out of the pursuit angle range, the advantage reward function will be negative and guide the UAVs to extend the distance from the enemy UAV. As the distance increases, the negative reward will increase to 0.

In addition, all UAVs on the winning side will receive 20 as an episode reward, and all UAVs on the losing team will receive -20 as an episode penalty.

Combine the three parts of the reward function to get the final UAV pursuit evasion reward function:

$$r_{j,t}^{(r)} = r_{j,t,catch}^{(r)} + r_{j,t,def}^{(r)} + r_{j,t,adv}^{(r)} \tag{19}$$

IV. Experiments

A. Experimental Setup

All experiments were run on a PC with Windows 11 (64-bit), an AMD Ryzen 9 processor, 64GB of memory, and an Nvidia GeForce RTX 3080 GPU. From a software perspective, we implemented the 15 UAVs v.s. 15 UAVs swarm pursuit-evasion environment (See Fig. 4 (a)) using the Multiagent Particle Environment (MPE) and implemented the algorithm using OpenAI Gym and the PyTorch library. To validate the performance of the proposed method, we selected baseline algorithms for comparison, including QMIX, MADDPG, and COMA.

B. Experimental Results

We train the proposed SMFMARL and the comparison algorithms for 40,000 episodes in the 15 UAVs v.s. 15 UAVs swarm pursuit-evasion environment, with each episode consisting of 100 timesteps. During the training process, we test the training agents after each update, collect the win rates, and show them in Fig. 2:

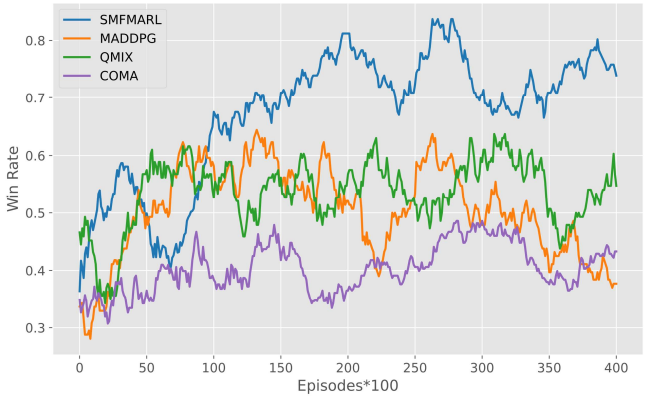

Fig. 2: Training results of win rates for different algorithms

The training results demonstrate that the proposed SMFMARL can effectively guide the UAV agents in the pursuit-evasion task and outperform the comparisons. The win rate curves fluctuate sharply because the parameters updated by the self-play agents are asynchronous with those of the training agents, which can decrease training difficulty but also introduce fluctuations.

To evaluate the performance of the proposed method, we compared the trained SFMARL algorithm with the well-trained comparisons and performed 200 episodes of simulation between algorithms, and show the win & loss rates of the SMFMARL algorithm in Fig. 3:

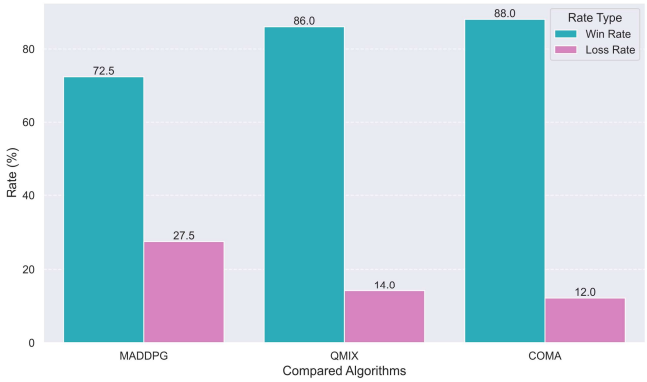

Fig. 3: The win & loss rates of different trained algorithms

Fig. 3 illustrates that the win rate of the SMFMARL algorithm is over 70%, which is the 3-times of the loss rate. Notably, the MADDPG algorithm performed worse in Fig. 2 but performed best in the evaluation stage, which indicates that the MADDPG is indeed in training. Overall, the evaluated win & loss rates results demonstrate that the proposed SMFMARL method outperforms the comparisons.

To validate the task performance of the proposed method, we present a case study of the well-trained SMFMARL algorithm in Fig. 4. In Fig. 4, the (a) - (i) are the screenshots from time 0-70 with a time interval of 10. First of all, from the perspective of tactical countermeasures, Fig. 4 (b)-(c) shows the middle part of the blue UAVs withdraws backward, forming an encirclement, while partial red UAVs, which will be caught by the concentrated firepower from the blue UAVs, gradually enter the encirclement. Meanwhile, when some of the red UAVs are eliminated, the red UAV swarm as a whole can retreat to avoid being further pursued by the blue UAVs and annihilated. Secondly, the red UAVs start their second pursuit in Fig. 4 (d)-(e), but they are still overwhelmingly caught by the blue UAVs, and they retreat again due to the concentrated firepower of the opposite side. In Fig. 4 (f)-(h), the blue UAVs choose to pursue actively, and multiple blue UAVs build up to a formation to encircle and suppress the remaining red UAVs. Finally, Fig. 4 (i) shows the scenario at $t = 75$, where all red UAVs are caught. The case study shows that the proposed method can guide the UAV agent to form a series of tactical strategies and complete the swarm pursuit-evasion task.

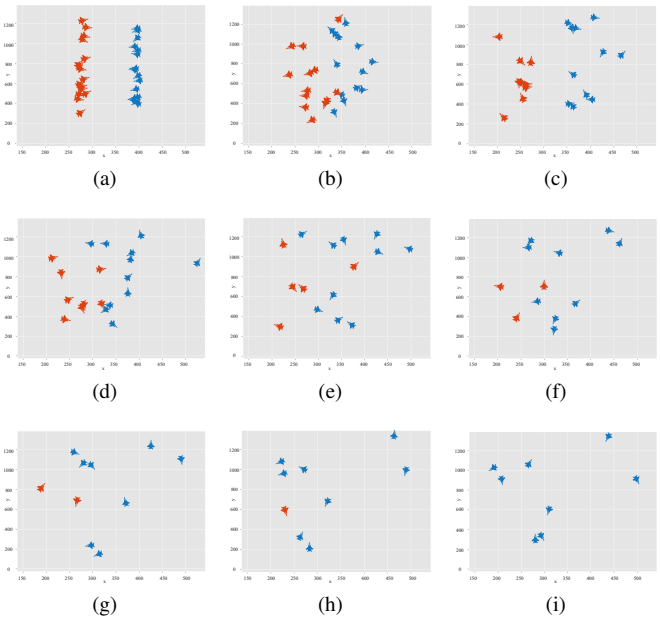

Fig. 4: The case study of SMFMARL pursuit evasion

To further demonstrate the tactics learned by the blue UAV swarms, screenshots of the confrontation at times 60, 61, and 62 are shown in Fig. 5 (a)–(c), respectively.

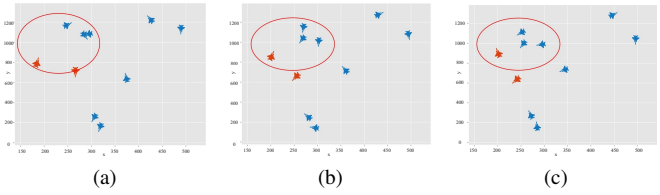

Fig. 5: The behavior study of SMFMARL pursuit evasion

As shown in Fig. 5 (a), the red circle included three blue UAVs, which already detected the red UAV swarm, but them are out of their pursuit range. In Fig. 5 (b) and (c), the blue UAV squads immediately adjust their direction, coordinating and moving towards the direction of the red UAVs, so that the red UAVs are included in the pursuit range. Finally, the blue UAV swarms caught the red UAVs. Overall, the results show the proposed method can learned effective countermeasures.

V. Conclusions

This paper introduced the Shapley Mean Field Multi-Agent Reinforcement Learning (SMFMARL) algorithm to address dimensionality explosion, credit assignment, and non-stationarity in UAV swarm pursuit-evasion tasks. By combining Approximate Marginal Contribution (AMC) for Shapley-Q value estimation with mean-field approximation to simplify inter-agent interactions, SMFMARL efficiently evaluates individual agent contributions while reducing computational complexity. A balanced reward function integrating catch, defensive, and advantage incentives enables UAVs to dynamically adapt to evolving scenarios and make autonomous decisions. Self-play and prioritized experience replay further enhance sample efficiency and policy robustness by reusing high-quality historical data to accelerate learning. Numerical simulations in a 15-UAV pursuit-evasion scenario demonstrate that SMFMARL outperforms existing MADRL algorithms in developing complex cooperative strategies.

References

[1] Z. Yang, Y. Cui, W. Du, F. Li, and Y. Li, "Cooperative pursuit-evasion with low altitude wireless network: A hierarchical reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 25, no. 4, pp. 5716–5729, 2026.

[2] L. Sun, H. Qiu, Y. Wang, and C. Yan, "Autonomous uav maneuvering decisions by refining opponent strategies," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 3, pp. 3454–3467, 2024.

[3] Z. Zhao, Y. Rao, H. Long, X. Sun, and Z. Liu, "Resource baseline mappo for multi-uav dog fighting," in *Proceedings of the 2021 International Conference on Autonomous Unmanned Systems, ICAUS*, 2021, pp. 3330–3336.

[4] Y. Guo, D. Xu, C. Wang, J. Li, and H. Long, "An invulnerable leader–follower collision-free unmanned aerial vehicle flocking system with attention-based multi-agent reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 160, p. 111797, 2025.

[5] R. Zhao, J. Tang, W. Zeng, Y. Guo, and X. Zhao, "Towards human-like questioning: Knowledge base question generation with bias-corrected reinforcement learning from human feedback," *Information Processing & Management*, vol. 62, no. 3, p. 104044, 2025.

[6] B. Wang, S. Li, X. Gao, and T. Xie, "Weighted mean field reinforcement learning for large-scale UAV swarm confrontation," *Applied Intelligence*, vol. 53, no. 5, pp. 5274–5289, 2023.

[7] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *Proceedings of the International Conference on Machine Learning, ICML*, 2018, pp. 5571–5580.

[8] X. Wang, L. Ke, G. Zhang, and D. Zhu, "Adaptive mean field multi-agent reinforcement learning," *Information Sciences*, vol. 669, p. 120560, 2024.

[9] M. Jusup, B. Pásztor, T. Janik, K. Zhang, F. Corman, A. Krause, and I. Bogunovic, "Safe model-based multi-agent mean-field reinforcement learning," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2024, pp. 973–982.

[10] H. Ma, Z. Pu, Y. Pan, B. Liu, J. Gao, and Z. Guo, "Causal mean field multi-agent reinforcement learning," in *International Joint Conference on Neural Networks, IJCNN*, 2023, pp. 1–8.

[11] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI*, 2018, pp. 2974–2982.

[12] J. Wang, Y. Zhang, T.-K. Kim, and Y. Gu, "Shapley q-value: A local reward approach to solve global reward games," in *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, 2020, pp. 7285–7292.

[13] J. Li, K. Kuang, B. Wang, F. Liu, L. Chen, F. Wu, and J. Xiao, "Shapley counterfactual credits for multi-agent reinforcement learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*, 2021, pp. 934–942.

[14] S. Han, H. Wang, S. Su, Y. Shi, and F. Miao, "Stable and efficient shapley value-based reward reallocation for multi-agent reinforcement learning of autonomous vehicles," in *Proceedings of the 2022 International Conference on Robotics and Automation, ICRA*, 2022, pp. 8765–8771.

[15] C. Wang, C. Yan, X. Xiang, and H. Zhou, "A continuous actor-critic reinforcement learning approach to flocking with fixed-wing uavs," in *Proceedings of the 11th Asian Conference on Machine Learning, ACML*, vol. 101, 2019, pp. 64–79.

[16] J. Wang, J. Wang, Y. Zhang, Y. Gu, and T. Kim, "SHAQ: incorporating shapley value theory into q-learning for multi-agent reinforcement learning," *CoRR*, vol. abs/2105.15013, 2021.

[17] S. Chen, Z. Zhang, Y. Yang, and Y. Du, "STAS: spatial-temporal return decomposition for solving sparse rewards problems in multi-agent reinforcement learning," in *Proceedings of the 38th AAAI Conference on Artificial Intelligence, AAAI*, 2024, pp. 17 337–17 345.