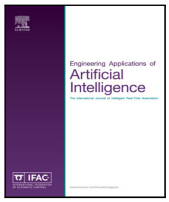
Engineering Applications of Artificial Intelligence 98 (2021) 104112

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Multi-agent hierarchical policy gradient for Air Combat Tactics emergence via self-play

Zhixiao Sun [a,1], Haiyin Piao [a,1,*], Zhen Yang [a,1], Yiyang Zhao [a,1], Guang Zhan [a,1], Deyun Zhou [a,1], Guanglei Meng [b,1], Hechang Chen [c,1], Xing Chen [c,1], Bohao Qu [c,1], Yuanjie Lu [d,1]

[a] Northwestern Polytechnical University, Xian, Shanxi, 710072, China
[b] Shenyang Aerospace University, Shenyang, Liaoning, 110036, China
[c] Jilin University, Changchun, Jilin, 130000, China
[d] Chinese Aeronautical Establishment, Beijing, 100012, China

ARTICLE INFO

Keywords:
Air combat
Artificial intelligence
Multi-agent reinforcement learning

ABSTRACT

Air-to-air confrontation has attracted wide attention from artificial intelligence scholars. However, in the complex air combat process, operational strategy selection depends heavily on aviation expert knowledge, which is usually expensive and difficult to obtain. Moreover, it is challenging to select optimal action sequences efficiently and accurately with existing methods, due to the high complexity of action selection when involving hybrid actions, e.g., discrete/continuous actions. In view of this, we propose a novel Multi-Agent Hierarchical Policy Gradient algorithm (MAHPG), which is capable of learning various strategies and transcending expert cognition by adversarial self-play learning. Besides, a hierarchical decision network is adopted to deal with the complicated and hybrid actions. It has a hierarchical decision-making ability similar to humankind, and thus, reduces the action ambiguity efficiently. Extensive experimental results demonstrate that the MAHPG outperforms the state-of-the-art air combat methods in terms of both defense and offense ability. Notably, it is discovered that the MAHPG has the ability of Air Combat Tactics Interplay Adaptation, and new operational strategies emerged that surpass the level of experts.

1. Introduction

Artificial Intelligence (AI) is a key technology affecting future air-to-air confrontation patterns (Byrnes, 2014). AI-driven combat aircraft will potentially capitalize on John Boyd's observe, orient, decide, action (OODA) loop, which will produce new and unmatched lethality to human-piloted aircraft (Osinga, 2007). They will also bring a revolution to air combat, as unmanned aircraft have shown much superiority, e.g., higher agility and harder overload durability. Therefore, AI air combat has often been a hot topic in aeronautical science research.

Researchers have paid their great efforts in rule-based methods (Burgin et al., 1975; Burgin and Eggleston, 1976; Burgin, 1976; Goodrich and McManus, 1989; Goodrich, 1993), probabilistic, fuzzy logic, computational intelligence methods (Virtanen et al., 2002, 2006; Ernest et al., 2016), machine learning, and reinforcement learning methods (McGrew et al., 2010; Kurniawan et al., 2019). Although significant progress has been made by these aforementioned approaches, they all rely heavily on prior human knowledge. For example, rule-based methods rely on pilots to pre-define air combat rule databases (Burgin et al., 1975). Probabilistic, fuzzy logic, and computational intelligence methods require experts to establish a probabilistic reasoning network or design a specific heuristic objective function (Virtanen et al., 2002; Ernest et al., 2016). Machine learning methods require a large number of data samples and usually need to be labeled by human experts (Floyd et al., 2017; Altan et al., 2018; Altan and Karasu, 2019; Altan and Hacıoğlu, 2020), and reinforcement learning methods rely on a per-step battlefield advantage function designed by human experts as dense reward (McGrew et al., 2010).

However, expert datasets are usually expensive and difficult to obtain. Even though datasets are available, they may impose a ceiling on the performance of systems trained in this manner (Silver et al., 2017). Moreover, in the real air combat process, actions involved are usually complicated (e.g., aiming, locking, firing, guidance, and avoiding missiles) and hybrid (e.g., discrete/continuous actions). It is infeasible to identify an optimal action sequence using rule-based methods considering expert knowledge, due to the large solution space caused by countless hybrid actions. For example, after selecting a discrete action, such as maneuver, the parameter determining a specific velocity is continuous and uncountable.

* Corresponding author.
E-mail address: haiyinpiao@mail.nwpu.edu.cn (H. Piao).
[1] All authors contributed equally.

https://doi.org/10.1016/j.engappai.2020.104112
Received 9 April 2020; Received in revised form 11 November 2020; Accepted 26 November 2020
Available online 7 December 2020
0952-1976/

In view of this, multi-agent reinforcement learning (MARL) algorithm is adopted to learn various strategies of air combat through adversarial self-play, to break the shackles of human expert knowledge. Moreover, hierarchical network is utilized to characterize the hybrid action decision-making problem, which can select complex action sequences efficiently and accurately. In recent years, MARL algorithms have demonstrated obvious superiority in computer games, e.g., Atari (Mnih et al., 2015), Dota2 (Berner et al., 2019), and StarCraft2 (Vinyals et al., 2019). Modern air-to-air combat is similar to the above problems, in terms of having a vast and dynamic solution space, assigning time credit for long-term sparse rewards, and requiring approximate real-time decision-making (Sutton and Barto, 2018). In summary, the contributions of this paper are as follows:

- A novel Multi-Agent Hierarchical Policy Gradient (MAHPG) algorithm is proposed, which enables the agent to exceed the cognition limitation of experts' prior knowledge by self-play. To the best of our knowledge, this is the first AI-enabled air combat agent with progressive tactics evolution ability.
- A hierarchical decision network is leveraged to select discrete/continuous actions efficiently. This gives the algorithm a hierarchical decision-making ability similar to that of humankind. Specifically, the first selected high-level discrete action (e.g., air combat maneuver) is transmitted to the subsequent continuous action (e.g., flight velocity command) prediction layer, so the overall action ambiguity have been effectively reduced.
- The effectiveness of the MAHPG is validated using an efficient air combat simulator named WUKONG[2] and comparing with four state-of-the-art deep reinforcement learning (DRL) methods and an expert-level rule-based bot. The MAHPG achieves the best comprehensive performance in terms of both defense and offense. Meanwhile, a phenomenon named Air Combat Tactics Interplay Adaptation (ATIA) is discovered (Fig. 1), and new operational strategies that surpass human experts emerged.

The remainder of this paper is organized as follows: After the literature review in Section 2, the proposed end-to-end MARL BVR (Beyond-Visual-Range) air combat training methodology is described in detail in Section 3. The experimental analysis is given in Section 4. Moreover, the ATIA phenomenon is discussed with the emerging process of multiple practical air combat tactics in Section 5. Finally, conclusions and future work are provided in Section 6.

2. Related work

In this section, previous studies about MARL, such as AlphaGo and AlphaGo Zero, will be introduced. This will be followed by AI-based methods in air combat like the rule-based, probabilistic, fuzzy logic, and computational intelligence methods.

2.1. Multi-agent reinforcement learning

Reinforcement learning (RL) considers single learning agents in a stationary environment. In contrast, MARL considers both collaborative and competitive multiple agents learning via RL, and usually, the non-stationarity introduced by other agents changes their behaviors as they learn (Hernandez-Leal et al., 2019). Recently, many MARL approaches have been proposed to scale to interactive decision-making problems that were previously intractable, i.e., settings with high-dimensional state and action spaces. Amongst recent work in the field of MARL, AlphaGo is one outstanding success story. This defeated a human world champion in Go (Silver et al., 2016), paralleling the historic achievement of IBM's Deep Blue in chess two decades earlier (Campbell et al., 2002). Unlike the hand-crafted rules that have dominated chess-playing systems, AlphaGo was composed of neural networks that were trained using supervised and reinforcement learning, in combination with a traditional heuristic search algorithm. A few months later, AlphaGo Zero was announced, which could defeat AlphaGo without any prior human knowledge and was trained from scratch using only self-play (Silver et al., 2017). After this, several breakthroughs in AI have been made in these domains by combining DRL with self-play, achieving superhuman performance. Challenging collaborative–competitive multi-agent environments have only recently been addressed using end-to-end MARL by Jaderberg et al. (2018) , which learns visually complex multi-player first-person video games to human level, as well as in continuous real-time domains, such as Dota2 (Berner et al., 2019) and Starcraft2 (Vinyals et al., 2019).

2.2. AI-based air combat

AI-based research in air combat dates to at least the 1970s. The National Aeronautics and Space Administration (NASA) first funded the development of a rule-based AI program for adaptive maneuvering logic (AML) to serve as a highly competent tactical driver in one-on-one air combat engagements. The AML generates elemental maneuvers, selects, and decides on the best maneuver, and delivers it to the execution control system to execute this maneuver (Burgin et al., 1975). In the late 1980s and early 1990s, as new technologies and capabilities were being proposed for agile, high-performance aircraft, the NASA Langley Research Center developed an integrated batch and piloted simulation tool known as the Tactical Guidance Research and Evaluation System (TGRES) (Goodrich and McManus, 1989; Goodrich, 1993). The TGRES evolved an advanced maneuvering logic that functions in real-time using AI and performs better than AML. However, rule-based systems suffer from the curse of dimensionality; when more conditions are confronted, more rules are needed to cover all potential situations. In the worst-case scenario, rules would have to be written for each combination.

The other important development in AI air combat is the use of probabilistic, fuzzy logic, and computational intelligence methods, including those of Virtanen's Influence Diagram approach (Virtanen et al., 2002, 2006) and Ernest's Genetic Fuzzy Tree (GFT) (Ernest et al., 2016). Furthermore, Floyd et al. (2017) described a Tactical Battle Manager (TBM) based AI technique to control an autonomous unmanned aerial vehicle in simulated BVR air combat scenarios (Floyd et al., 2017). Ramirez et al. developed domain-independent planners embedded into professional multi-agent simulations, to implement two-level Model Predictive Control (MPC) hybrid control systems for simulated UAVs (Ramirez et al., 2018). These methods first establish a preliminary reasoning topological structure by humans. They then involve Bayesian reasoning or genetic algorithms to obtain an optimized or evolved topology. Although, as experiments have shown, these methods have defeated some human pilots (Ernest et al., 2016), the establishment of a preliminary reasoning structure still depends on human effort, so some limitations still exist.

Recently, RL-based AI air combat algorithms present the potential of simplifying the development and maintenance of complex autonomous systems by learning optimal behaviors from continuous simulations. Such learning can be done in an unsupervised manner and simply needs processor time during learning. Toward this particular air combat AI solution, McGrew used approximate dynamic programming to solve a fixed velocity, one-on-one air combat maneuvering problem in a two-dimensional space (McGrew et al., 2010). In addition, Kurniawan et al. (2019) proposed an empirical study of reward structures for actor–critic RL in air combat maneuvering simulations.

All the aforementioned works that used RL focused on human expert-designed dense reward approaches. In contrast, our research is significantly different as it trains two adversarial air combat agents with solely self-play in an end-to-end MARL manner with highly sparse and objective key event-based rewards. All training results are accomplished without any human handcrafted rules. Moreover, this approach induces a sequence of challenges for the adaptive process that we term ATIA, which is proven to develop many complex and interesting air combat tactical behaviors progressively.

[2] WUKONG means "Zen of Aerospace" in Chinese.

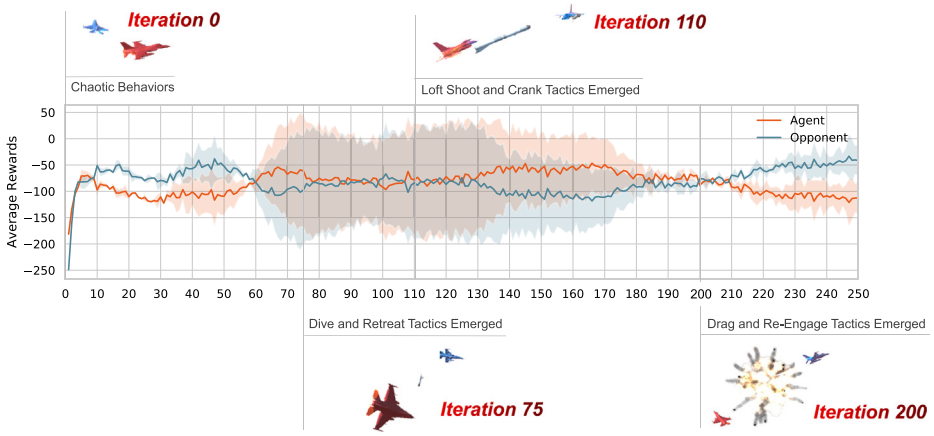


Fig. 1. Air combat tactics emergence via ATIA - The red and blue curves denote the training rewards gained by two opposite agents. The shaded region denotes a standard deviation of average evaluations over three trials. The emerged tactics can be classified into three main categories: (1) Dive and retreat: Continuously lower down the altitude and turn 180° to retreat. (2) Loft shoot and crank: Shoot after pulling the aircraft nose up for achieving longer missile attack range, then horizontally turn 50° and keep the enemy in track with the radar. (3) Drag and re-engage: Horizontally turn 90° for breaking away from incoming missiles, then turn back to attack at an appropriate time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Methodology

In this section, the beyond-visual-range (BVR) air combat problem is first formulated as a Markov game. Then, an end-to-end MARL approach for training an air combat agent from scratch, with the proposed MAHPG algorithm and its neural network architecture described in detail, is discussed. Finally, a key event-driven reward-shaping (KAERS) mechanism is proposed.

3.1. Problem formulation

One-on-one BVR air combat involves controlling one player during combat against an enemy player. Therefore, each air combat scenario can be considered as a competitive game between two adversarial agents. In this paper, we formulate it as a multi-agent Markov Decision Processes (MDPs) called Markov games (Littman, 1994).

A Markov game for N agents is defined by a set of states $\{S^i\}_N^{i=0}$ and a set of joint actions $\{A^i\}_N^{i=0}$ for each agent. Each agent i observes a private system state S^i and selects actions through its policy $\pi^i : S^i \times A^i \to [0,1]$, producing the next state according to the state transition function $P : S \times A^1 \times \cdots \times A^N \times S \to [0,1]$. Each agent i aims to maximize its won total expected sum of rewards:

$$R_t^i = E\left\{\sum_T^{t=0} \gamma^t r_t^i\right\} \tag{1}$$

where γ is a discount factor, T is the time horizon, and r_t^i is the reward received t steps of agent i. The agents' joint policy induces a state-value function, i.e., an expectation over R_t, $V^{\pi_i}(s_t) = E\left[R_t \mid s_t\right]$, while the action-value function is defined as $Q^{\pi_i}(s_t, a_t^i) = E\left[R_t^i \mid s_t, a_t^i\right]$. The advantage function $A^{\pi_i}(s_t; a_t^i) = Q^{\pi_i}(s_t, a_t^i) - V^{\pi_i}(s_t)$ describes whether taking action a_t^i is better or worse for agent i when in state s_t than the average action of policy π_i, i.e., V^{π_i}.

A Markov game is called a two player zero-sum Markov game when both players are fully-competitive and the sum of rewards is zero for these agents. To solve one-on-one simulated air combat games in this specific theoretical framework, a problem mapping mechanism is needed for converting physical air combat problems into formalized Markov games.

3.2. Air combat states, actions, rewards, and objectives

In this section, the one-on-one BVR air combat problem is mapped to a Markov game by specifying air combat states, actions, rewards as well as objectives.

For system states, beyond the aircraft's global position, altitude, velocity, and normal load factor states, to be successful in air combat, the agent's aircraft needs to be in a specific relative aircraft geometry with the opponent. McGrew's geometry annotation is adopted as a baseline (McGrew et al., 2010), and define the relative state observation calculation, and the relative state observation calculation is defined. The aircraft centers of mass are connected by the line of sight (LOS) line, which is also used to calculate the range between the two aircrafts. The aspect angle (AA) is the angle between the LOS line and the tail of the opponent aircraft. The antenna train angle (ATA) is the angle between the nose of the agent aircraft and the LOS line. Both the AA and ATA help the pilot to make maneuvering decisions. By convention, angles to the right side of the aircraft are considered positive, and angles to the left are considered negative. Then the radar, radar warning receiver (RWR), and mid-range missile status are included at the end of the whole state vector, and the total air combat state definition can be described as follows (refer to Appendix C for more details):

$$\begin{aligned} s^i = \big[&x, x_b, v, v_b, \psi, \theta, \phi, n_n, r, \dot{r}, AA, \\ &ATA, EL, \Delta h, lo, warn, m_{left}, T_{go}\big]^T \end{aligned} \tag{2}$$

Furthermore, 14 offensive and defensive basic fighter maneuver (BFM) macro actions were designed for this particular full BVR game, as summarized in Appendix D. Basic fighter maneuvers have been described as the art of maneuvering a combat aircraft to obtain a position from which an attack can be made on another aircraft and representing the primary elements that can be viewed as the building blocks for air combat maneuvers. These are composed of accelerations/decelerations, climbs/descents, and turns that can be performed in combination relative to other aircrafts. Therefore, the total combat strategy consists of the composition or series of atomic BFM operations. To ensure the maneuverability, the agent needs to determine the continuous normal load factor command n_{nc} and the velocity command v_c, correspondingly, after choosing a discrete optimal maneuver under the current situation s. This makes air combat extend to a discrete/continuous hybrid action space problem. With such an abstraction in the action space, it is easier for agents to learn a high-level strategy with human-level flexibility of air combat micro-operations.

As previous works relied on per-step dense reward signals tuned by human experts (McGrew et al., 2010; Kurniawan et al., 2019), a method is proposed to solve the prohibitively hard credit assignment problem of learning from sparse and delayed episodic win/loss signals, optimizing hundreds of actions based on a single final reward with some sparse and objective key events (see Table 1). It is believed that this assumption

Table 1
Key air combat event reward shaping (KAERS).

Categories	Event name	Weights	Description
Episodic rewards	Kill	+500	Opponent's aircraft is shot down by the chasing missile
	Be killed	−500	Agent's aircraft is shot down by missile launched from enemy
	Crash	−500	The height of agent's aircraft falls below zero
	Drove out	+500	Agent's aircraft drives opponent out of border more than 60 s
	Be driven out	−500	Agent's aircraft is driven out of border more than 60 s
Event based rewards	Shoot	−25	Agent's aircraft launches a missile
	Stall	−50	Agent's aircraft remains a high angle of attack and keep on losing height
	Out	−10	Agent's aircraft is out of battlefield border
	Lock on	+2	Agent's aircraft successfully locks on its opponent
	Be locked	−2	Agent's aircraft is locked by enemy's radar
	Missile lock	+5	The chasing missile's seeker locks its opponent
	Missile alert	−5	Agent's aircraft is locked by opponent's missile
	Lock escaped	+5	Agent's aircraft escapes from opponent's radar tracking
	Missile escaped	+10	Agent's aircraft escaped from opponent's missile tracking

will lead to an unbiased solution to the true winning policy rather than over-fitted to some human expert-crafted potential functions. In these solutions, the causal relationship with the true combat result is mathematically ambiguous.

The agent's objective is to learn a policy that maximizes the expected sum of discounted rewards. Conversely, the opponent's joint policy is to minimize the expected sum. Correspondingly, we have the following mini-max zero-sum Markov game:

$$Q^*\left(s_t, a_t^i, a_t^{-i}\right) = r\left(s_t, a_t^i, a_t^{-i}\right) + \max_{a^i \in \pi^i} \min_{a^{-i} \in \pi^{-i}} Q^*\left(s_{t+1}, \left\langle a_{t+1}^i, a_{t+1}^{-i} \right\rangle\right) \tag{3}$$

where $Q^*\left(s_t, a_t^i, a_t^{-i}\right)$ is the optimal action-state value function, which follows the Bellman Optimal Equation. For solving such kinds of games, agent policies are trained against each other by self-play, which means fictitious players choose the best responses to their opponents' average behavior. The average strategies of fictitious players were proven to approximate to the $\epsilon-$Nash Equilibrium in this particular kind of games (Leslie and Collins, 2006; Heinrich et al., 2015).

3.3. Multi-agent hierarchical policy gradient (MAHPG)

In an actual air combat process, human pilots usually first choose a reasonable discrete BFM under the current situation. Then, in the process of executing this maneuver, they must consistently adjust the continuous normal load factor command. Then, in the process of executing this maneuver, they have to consistently adjust the continuous normal load factor command n_{nc} and the velocity command v_c through the coordinated control of throttle and stick, which naturally forms a discrete/continuous hybrid hierarchical decision-making structure. To address this problem, a MAHPG algorithm was proposed, utilizing the hierarchical air combat decision-making characteristics. Particularly, the following hybrid action space was adopted: the high-level discrete maneuver is selected from a finite set $M_d = \{m_1, m_2, \ldots, m_k\}$, and each specified maneuver $m \in M_d$ contains a continuous parameter set $X_m = \{n_{nc}, v_c\}$. In this way, a complete hybrid air combat action is represented as $a = [m, f, n_{nc}, v_c]^T$, where f is a boolean action parallel to high level maneuver action m indicates the missile firing signal.

The multi-agent hierarchical actor–critic architecture is adopted and improved, and the proposed network structure for a specified agent in the multi-agent settings consists of three parts: (1) the high level discrete maneuver and shoot decision network π_θ utilizes an exponential softmax distribution $\pi_\theta(a|s) = \frac{e_\theta^{h(s,a)}}{\sum_b e_\theta^{h(s,b)}}$ as outputs, where $h(s,a)$ and $h(s,b)$ indicates the previous layer output logits. In such form, the overall high level policy network's outputs of the chosen maneuver $p(m)$ and firing signal $p(f)$ are treated as two independent probability distributions, but sharing the same hidden layers for modeling as an inner joint distribution, as described in Eq. (4)–(5). (2) The low level continuous normal load factor command n_{nc} and steering velocity v_c command decision network π_ϕ adopts diagonal Gaussian distribution to fit the normalized mean value μ and the standard deviation σ, respectively. (3) The centralized value network takes states from all agents as its input. In which θ, ϕ, and φ are trainable parameters for the three neural networks. All these networks consist of two cascading full-connection hidden layer with 512/256 nodes and each layer adopts $ReLU$ as the activation function, as shown in Fig. 2.

In the entire maneuver decision-making process, which is determined by two-layer cascading neural networks, the appropriate maneuver m and the fire signal f are inferred simultaneously by using the high level discrete maneuver selection network $\pi_\theta(m, f|s)$. Then, the observed state s and the inferred maneuver m are taken as the input of the low level neural network. As a result, n_{nc} and v_c commands are obtained. To make full use of the hierarchical maneuvering decision nature of air combat, the overall policy is represented in the chain rule as described in Eq. (6). This representation is arguably simpler as it interprets the continuous commands fitting problem as a posterior distribution of $\pi_\phi\left(n_{nc}, v_c|s\right)$. And the $\pi_\theta(m|s)$ output can be interpreted as prior knowledge, thus transforms the problem of choosing a complicated full discrete/continuous hybrid action $[m, f, n_{nc}, v_c]^T$ into an easier cascading neural networks prediction sequences.

$$m, f \sim \pi_\theta(m, f|s) \tag{4}$$

$$\pi_\theta(m, f|s) = \pi_\theta(m|s) \cdot \pi_\theta(f|s) \tag{5}$$

$$n_{nc}, v_c \sim \pi_\phi\left(n_{nc}, v_c|m, s\right) \tag{6}$$

Considering that all agents hold some key hidden states during air combat process, which will significantly affect the battle situation judgment correctness, e.g., the missile launch signal. However, if the opponent could see the accurate enemy missile launch signal during training, the agent can obviously evaluate the current air combat situation more accurately thus making more correct decisions. Formally speaking, if some key states remain invisible as the episode processes, the problem will degenerate from MDP to POMDP (Partially-Observed Markov Decision Process). In such setting, the V^{π_i} is biased from the ground truth. Please refer to Appendix B for more details. For solving this problem, we proposed a centralized global value function architecture for each agent, which has directly access to the full environment state without any imperfect information due to visibility, as described in Eq. (7). Where φ represents the value network parameters, s_{all} represents the concatenated state variables for all agents in the air combat scenario. By sharing fully observed states from all agents, the global visibility of the state can be ensured directly, therefore, the estimation error of V_{tot}^i is reduced.

$$V_{\varphi,tot}^i = f\left(s_1, s_2, \ldots, s_n\right) = f\left(s_{all}\right) \tag{7}$$

In the neural networks training process, the parameters of the hierarchical hybrid policy networks are learned concurrently by Proximal Policy Optimization (PPO) (Schulman et al., 2017) as a joint optimization. For each agent in the air combat scenario, this joint optimization maximizes the unified accumulated reward expectation with updating π_θ and π_ϕ collaboratively. This maximization process generates two separate policy gradients, and the generated policy gradients aim to

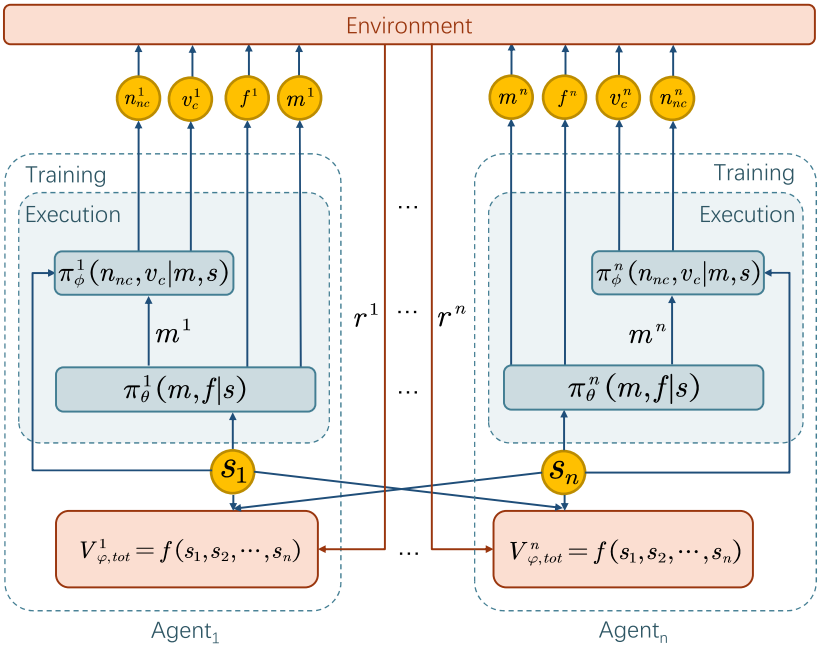


Fig. 2. Air combat multi-agent neural network architecture.

estimate the gradient of unified expected returns V_{tot}^{i} with respect to the parameters of the discrete maneuver and shoot decision policy:

$$\nabla_{\theta}\mathcal{J}\left(\pi_{\theta}\right)=E_{m,f\sim\pi_{\theta}}\left[\nabla_{\theta}\log\left(\pi_{\theta}\left(m_{t}^{i},f_{t}^{i}|s_{all,t}\right)\right)A_{\varphi,tot}^{i}\left(s_{all,t},a_{t}^{i}\right)\right] \tag{8}$$

And the subsequent continuous steering parameters n_{nc} and v_c fitting policy:

$$\nabla_{\phi}\mathcal{J}\left(\pi_{\phi}\right)=E_{n_{nc},v_{c}\sim\pi_{\phi}}\left[\nabla_{\phi}\log\left(\pi_{\phi}\left(n_{nc,t}^{i},v_{c,t}^{i}|s_{all,t}\right)\right)A_{\varphi,tot}^{i}\left(s_{all,t},a_{t}^{i}\right)\right] \tag{9}$$

where $\mathcal{J}(\pi_{\theta})$ and $\mathcal{J}(\pi_{\phi})$ are the policy gradient manner corresponding accumulated expected returns. Here, PPO is adopted for the generated policy gradients to penalize large changes to the policies for preventing training instabilities as a clipped surrogate objective function. The multi-agent hierarchical hybrid policy networks parameters θ and ϕ could be updated according to the gradient of the proposed hierarchical PPO loss functions. Specifically, we use Q-value function Q, which gives the expected sum of remaining rewards received starting from $s_{all,t}, a_{t}^{i}$, and advantage A, which can be considered as another version of Q-value with lower variance, by taking the state-value off as the baseline, as ordinary setting:

$$Q_{tot}^{i}\left(s_{all,t},a_{t}^{i}\right)=E\left[r_{t+1}^{i}+\gamma V_{\varphi,tot}^{i}\left(s_{all,t+1}\right)\right] \tag{10}$$

$$A_{\varphi,tot}^{i}\left(s_{all,t},a_{t}^{i}\right)=Q_{\varphi,tot}^{i}\left(s_{all,t},a_{t}^{i}\right)-V_{\varphi,tot}^{i}\left(s_{all,t}\right) \tag{11}$$

The objective functions, which are variants of the expected return, are directly used in the form as PPO claimed:

$$\begin{aligned}\mathcal{J}\left(\pi_{\theta}\right)=&E_{m,f\sim\pi_{\theta}}\left[\min\left(l_{t}\left(\theta\right)A_{\varphi,tot}^{i},\right.\right.\\&\left.\left.clip\left(l_{t}\left(\theta\right),1-\varepsilon,1+\varepsilon\right)A_{\varphi,tot}^{i}\right)\right]\\&-E_{s\sim\pi_{\theta}}\left[\alpha H\left(s_{all,t},\theta\right)\right]\end{aligned} \tag{12}$$

$$\begin{aligned}\mathcal{J}\left(\pi_{\phi}\right)=&E_{n_{nc},v_{c}\sim\pi_{\phi}}\left[\min\left(l_{t}\left(\phi\right)A_{\varphi,tot}^{i},\right.\right.\\&\left.\left.clip\left(l_{t}\left(\phi\right),1-\varepsilon,1+\varepsilon\right)A_{\varphi,tot}^{i}\right)\right]\\&-E_{s\sim\pi_{\phi}}\left[\alpha H\left(s_{all,t},\phi\right)\right]\end{aligned} \tag{13}$$

The importance sampling, which is often used to correct the mismatch between behavior policy π_{old} and target policy π_{θ}, is in the form as:

$$l_{t}\left(\theta\right)=\frac{\pi_{\theta}\left(m_{t}^{i},f_{t}^{i}\mid s_{all,t}\right)}{\pi_{old}\left(m_{t}^{i},f_{t}^{i}\mid s_{all,t}\right)} \tag{14}$$

$$l_{t}\left(\phi\right)=\frac{\pi_{\phi}\left(n_{nc,t}^{i},v_{c,t}^{i}\mid s_{all,t},m_{t}^{i}\right)}{\pi_{old}\left(n_{nc,t}^{i},v_{c,t}^{i}\mid s_{all,t},m_{t}^{i}\right)} \tag{15}$$

where both l_t denote the likelihood ratio between new and old policy hierarchies and $clip\left(l_t,1-\varepsilon,1+\varepsilon\right)$ clips $l_t(\theta)$ and $l_t(\phi)$ in the interval $[1-\varepsilon,1+\varepsilon]$. $\alpha H\left(s_{all,t},\theta\right)$ and $\alpha H\left(s_{all,t},\phi\right)$ are entropy regularization penalty functions for encouraging policy exploration and α is the scaling factor. The hierarchical hybrid policies and the centralized value function $\mathcal{L}\left(V_{\varphi}\right)$ are adjusted toward a look ahead value, thus the centralized value function loss can be defined and the value network can be updated.

$$\mathcal{L}\left(V_{\varphi}\right)=E_{s_{all}\sim\pi_{\theta},\pi_{\phi}}\left[\left(\sum_{j=0}^{n-1}r_{t+j}+\gamma^{n}V_{\varphi,tot}^{i}\left(s_{all,t+1}\right)-V_{\varphi,tot}^{i}\left(s_{all,t}\right)\right)^{2}\right] \tag{16}$$

Consequently, the overall proposed method of Multi-Agent Hybrid Action Space Policy Gradient (MAHPG) is described in Algorithm 1.

3.4. Key air combat event reward shaping

The main purpose of one-on-one BVR air combat is to kill or drive out opponents within a specified time. Effective killing depends on a series of precise operations, including aiming, locking, firing, guidance, and avoiding missiles fired by an enemy (Bonanni, 1993). In the case of only a win/lose signal as an episodic reward, the probability of these actions series occurring simultaneously is obviously close to zero. However, the success of agents in the competitive games requires the agents to occasionally solve the task (i.e., win the competition) by random actions. In MARL theory, this kind of problem is referred to as long-term sparse reward time credit assignment. In such a scenario, since the neural network of the agent is initialized randomly by Xavior method (Glorot and Bengio, 2010), the lack of any basic skills related to air combat at the very beginning could seriously impact the cold

Algorithm 1: Multi-Agent Hybrid Action Space Policy Gradient (MAHPG)

for $iteration = 1, \cdots, N$ **do**
  for $agent = 1, \cdots, n$ **do**
    for $t \in \{1, \cdots, T\}$ *timesteps* **do**
      Use high level discrete policy π_θ to sample m and f
      Feed both s_{all} and m into low level continuous policy π_ϕ to sample n_{nc} and v_c
      Step in simulation environment
      Store per-agent's transition $\left\{ s_t^i, m_t^i, f_t^i, n_{nc,t}^i, v_{c,t}^i, r_{t+1}^i, s_{t+1}^i \right\}$ separately
  Estimate Advantages $A_{\varphi,tot}$
  for $agent = 1, \cdots, n$ **do**
    for *update hierarchical hybrid policies' parameters M times* **do**
      Calculate $\mathcal{J}\left(\pi_\theta\right) = E_{m,f\sim\pi_\theta}\left[\min\left(l_t\left(\theta\right)A_{\varphi,tot}^i, clip\left(l_t\left(\theta\right), 1-\varepsilon, 1+\varepsilon\right)A_{\varphi,tot}^i\right)\right] - E_{s\sim\pi_\theta}\left[\alpha H\left(s_{all,t}, \theta\right)\right]$
      Update high level policy parameters θ by gradient descending w.r.t. $\mathcal{J}\left(\pi_\theta\right)$
      Calculate $\mathcal{J}\left(\pi_\phi\right) = E_{n_{nc},v_c\sim\pi_\phi}\left[\min\left(l_t\left(\phi\right)A_{\varphi,tot}^i, clip\left(l_t\left(\phi\right), 1-\varepsilon, 1+\varepsilon\right)A_{\varphi,tot}^i\right)\right] - E_{s\sim\pi_\phi}\left[\alpha H\left(s_{all,t}, \phi\right)\right]$
      Update low level policy parameters ϕ by gradient descending w.r.t. $\mathcal{J}\left(\pi_\phi\right)$
    for *update value net parameters B times* **do**
      Calculate $\mathcal{L}\left(V_\varphi\right) = E_{s_{all}\sim\pi_\theta,\pi_\phi}\left[\left(\sum_{j=0}^{n-1} r_{t+j} + \gamma^n V_{tot,t}^i\left(s_{all,t+1}, \varphi\right) - V_{tot,t}^i\left(s_{all,t}, \varphi\right)\right)^2\right]$
      Update value net parameters φ by gradient descending w.r.t. $\mathcal{L}\left(V_\varphi\right)$

start problem (Ding and Soricut, 2017). To address this problem, the Key Air Combat Event Reward Shaping (KAERS) method is proposed. The main purpose is to provide relatively sufficient signal stimulation at the beginning of the agents' training by discrete key events during air combat, as described in Table 1. These eventually shaped rewards can initially be used to allow the agents to learn basic air combat skills.

$$r_t = \lambda e_t^T w + r_{episodic} \tag{17}$$

As described in Table 1, r_t is the t step total reward, e_t is a key event, and w is the reward scale weight according to the corresponding event. The key event shaped reward is gradually annealed to zero linearly after the 100th iteration, in favor of the true end game episodic reward $r_{episodic}$, to allow the agents to train for the majority of the time using the sparse episodic reward. This is achieved using a linear annealing factor of λ. Follow-up experiments show that the KAERS method can effectively address the cold start problem in the early stages of air combat agent training, thereby sufficiently accelerating the total training process.

3.5. Training process

In the self-play training process of the MAHPG algorithm, we initialize all agents' neural networks by the Xavier initialization method to produce brand new networks (Glorot and Bengio, 2010), which means that no human air combat knowledge is adopted by MAHPG. In the adversarial training scenario, we truncate the trajectory and calculate the advantage function $A(s,a)$ after $n = 30$ forward steps of a network or if a terminal signal is received. The optimization process runs 16 asynchronous processes using a shared Adam numerical optimizer. For each parallel process, we ran experiments until the total episode replay buffer (Mnih et al., 2015) gathered a specified batch size of samples, where the learning rate is 1.0×10^{-4}. We use an independent entropy penalty of 4.0×10^{-5} for the action heads. All the above training hyperparameters come from expert tuned values adopted from Mnih et al. (2016), Lowe et al. (2017) and Berner et al. (2019). The environment performs a fixed updating rate as one step per second which involves a balance between simulation performance and precision. For action consistency consideration as introduced in Mnih et al. (2015), we act every four game steps, which is equivalent to four seconds per action.

4. Experiments

In this section, the simulation environment for air combat will be introduced first. Then, experimental results against the state-of-the-art will be given and analyzed. Finally, the effectiveness of the proposed method compared with an expert-level bot will be evaluated.

4.1. WUKONG: Air combat simulator

The experimental environment used in this paper is WUKONG, a MARL-oriented air combat simulation framework currently under development by Northwestern Polytechnical University (NPU). WUKONG is designed to simulate teams of aircraft in both BVR and Within-Visual-Range (WVR) adversarial n-versus-m air combat. The environment is designed for air combat operation and AI-driven combat behavior research.

The scenario considered in this paper consists of one red and one blue fighter[3] jet, each comprising of a single entity representing an aircraft. Each aircraft consists of four components: aircraft flight dynamics, radar and RWR, mid-range missiles, and an expert-level built-in AI bot. The purpose of our BVR air combat scenario is to shoot the enemy down or to drive them away. The environment also offers the state of both aircraft's radar/RWR sights, which includes information on the orientation of the entities list that the radar/RWR has detected and could track. The radar target searching procedure is not discussed as perfect observation of the opponent's states is being assumed. Thus, the radar is only used for locking on the opponent before a shot and performing missile mid-guidance procedures.

4.2. Comparison with baseline algorithms

To show the learning performance of MAHPG, we compare the performance of the proposed MAHPG algorithm with four state-of-the-art RL/MARL-based air combat algorithms: (1) P-DQN, (2) MADDPG, (3) PPO with win/lose signal, and (4) A2C with McGrew score. The McGrew score is the widely used expert-crafted air combat shaped rewards (McGrew et al., 2010); P-DQN is a baseline RL solution for hybrid action space problems (Xiong et al., 2018); MADDPG is an

[3] Also marked as agent and opponent for convenience.

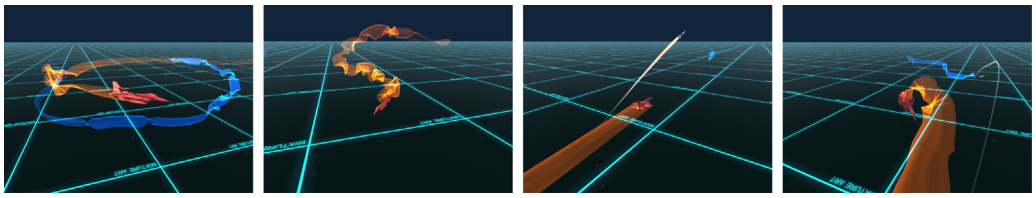


Fig. 3. Learned behaviors among algorithms during training. (a) Approaching an opponent's 6 o'clock aspect. (b) Conservative spinning around. (c) Fire at an effective distance. (d) Hierarchical hybrid maneuvers act more agile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

effective MARL baseline; and PPO and A2C are state-of-the-art RL algorithms (Lowe et al., 2017). Algorithms (1)–(3) adopt KAERS mechanism as reward functions. It should be noted that algorithms (2)–(4) cannot deal with discrete/continuous hybrid action space originally. Hence, these algorithms were used to choose high-level maneuver action m and shoot action f only, and determine the corresponding continuous n_{nc} and v_c parameters with the fixed values designed by human experts as described in Table 6.

The experiment is set to collect 960,000 self-play samples for training five independent air combat agents. Then the average training rewards are revealed, and the battle result statistics between MAHPG and other algorithms are determined. The training rewards comparison results are shown in Fig. 4 (the shaded region denotes a standard deviation of average evaluations over five trials). With the training procedure processed, all algorithms achieved relatively high scores from large negative starting points, which demonstrated that all algorithms had learned some sort of air combat knowledge solely from self-play. Comparing with the other four baselines, the initial learning speed of MAHPG performs better than P-DQN and MADDPG during training but falls lower when the training process ends. PPO with win/lose signal performs much better than the others, and A2C with McGraw score performed mediocrely.

Since the McGrew score is a dense human-crafted per-step shaped reward mechanism, it is easier to obtain higher training rewards. From air combat replay visualization (Fig. 3(a)), it was observed that the agent trained by the A2C with the McGrew score quickly converged to approaching an opponent's 6 o'clock aspect, which was exactly consistent with the McGraw score principle. Therefore, it could be determined that although this approach gained more rewards, it is also over-fitted to some kind of human-crafted strategy. However, the PPO with the win/lose signal still achieve higher scores than MAHPG. By replay visualization, it was revealed that in the early stage of training, the PPO with the win/lose signal only learned a conservative spinning around strategy (Fig. 3(b)). Such behaviors ensure that both sides gain approximate 0 scores, but obviously, they are not able to discover new air combat tactics. In the later period of training, as the adversaries slowly began to launch some missiles randomly with enemy killed after mastering simple flight skills, a larger reward variance was induced, as shown in Fig. 4.

Although the MAHPG and P-DQN methods gained lower training rewards than the PPO with the win/lose signal and A2C with the McGrew score, from the replay visualization, it can be observed that the agents trained by MAHPG and P-DQN had chosen discrete tactical maneuvers wisely and exhibited a variety of continuous operation commands according to the current situation, performing more agile and in a maneuverable way (Fig. 3(d)). The training score of the MAHPG descending trend (Fig. 4) reveals that the MAHPG has learned to fire at an effective distance (Fig. 3(c)) and started to evade incoming missiles before other algorithms. Also, the training score declined reasonably due to the increasing probability of effective shots.

Furthermore, MAHPG trained with the KAERS mechanism encouraged the willingness of adversaries to fight each other at the early stages of training. For example, a successful radar lock on operation gained a +2 score, which drove the agent to lock opponents with more incentive. Moreover, since successful radar lock-on is the premise of a missile launch, it increases the probability of successful shots, which produces more opportunities for adversaries to discover the necessity of evading incoming missiles launched by opponents in the early stages.

Table 2
Effective shots and evades statistics trained with 480,000 and 960,000 samples in 30 air combat confrontations.

Algorithm	Samples = 480,000		Samples = 960,000	
	Effective shots	Evades	Effective shots	Evades
MAHPG	16	14	33	25
P-DQN	11	9	16	14
MADDPG	8	6	15	9
PPO with win/lose signal	13	11	27	16
A2C with McGrew score	7	5	10	6

Table 3
Binomial test between MAHPG and baselines.

Algorithm compared	Win times	Lose times	p-value
P-DQN	49	34	3.9×10^{-2}
MADDPG	44	31	5.3×10^{-2}
PPO with win/lose signal	60	18	2.0×10^{-7}
A2C with McGrew score	52	10	5.0×10^{-9}

In order to ascertain this conclusion, effective shots (shots with $T_{go} < 25$ s) and evades (defensive maneuvers adopted with incoming missile's $T_{go} < 20$ s) were counted in 30 air combat simulations after 480,000 and 960,000 samples collected and trained among algorithms in Table 2. The results conclusively demonstrated that MAHPG trained with KAERS mechanism enhanced both shot and evade behaviors better than the other algorithms. Although the training score of MAHPG is not the highest, it performs better than other methods in terms of behavior emergence and air combat knowledge discovery.

To prove this conclusion, 100 air combat battles were carried out between agents trained by the MAHPG and the others after training had ended. As shown in Fig. 5, from the air combat match results, it can be clearly seen that our method achieved the highest winning rate.

To further determine this, a binomial hypotheses test was run on the battle statistics between MAHPG and other baseline algorithms. For eliminating the impact of uncertainty induced by the draw game, the test excluded all draw games. Since we claim that MAHPG can defeat all other algorithms, the null hypothesis was declared as MAHPG will achieve lower than 50% winning rate against all others. As shown in Table 3, the null hypothesis should be rejected at the 6% level of significance because all returned p-values are lower than the criterion. Therefore, the statistical hypothesis test result also proves the effectiveness of MAHPG.

4.3. Comparison with expert-level bot

To demonstrate that the MAHPG can also defeat expert-level opponents beyond scoring, it was arranged to fight with the built-in hand-crafted bot in the WUKONG environment every 20 iterations as training progressed until 200 iterations had ended (refer to Fig. 6).

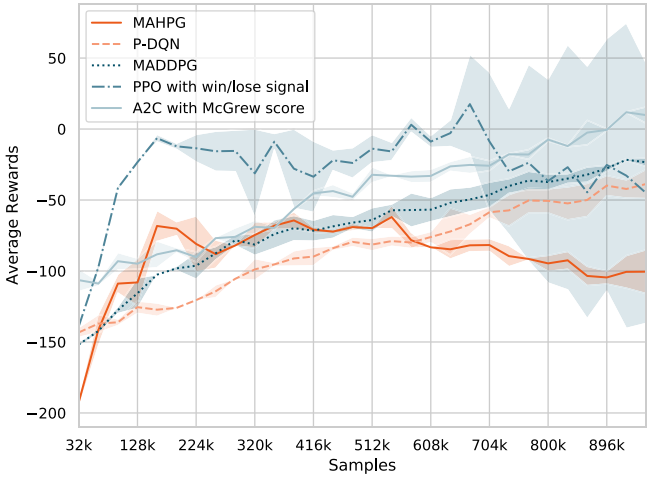


Fig. 4. Training rewards comparison.

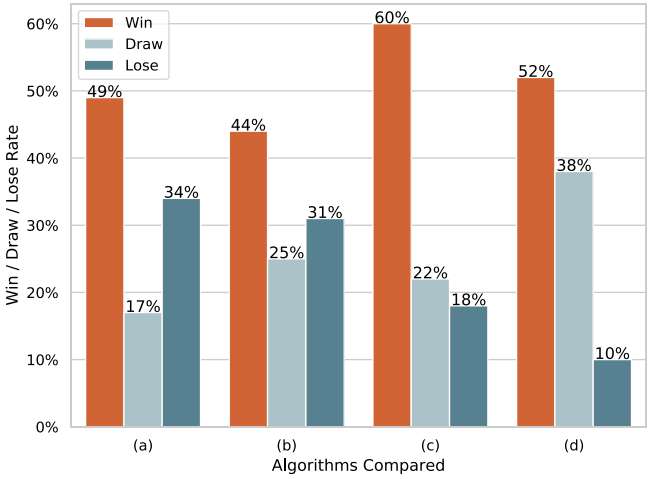


Fig. 5. Air combat battle results between algorithms. (a) vs. P-DQN, (b) vs. MADDPG, (c) vs. PPO with win/lose signal, and (d) vs. A2C with McGrew score.

The bot was modeled by an expert-level rule-based BVR air combat state machine based on jet pilot knowledge provided by air combat manuals (Bonanni, 1993; Shaw, 1985). This performs complete engagement, e.g., target search, lock, shoot, and defense tactics. Since the bot's policy cannot evolve, the MAHPG can gradually beat the benchmark expert opponents through training. As can be seen, when training started, the average win rate of MAHPG was only 0% while the bot's rate was as high as 100%. By the end of the 200 iterations, our algorithm reached 68%, while the bot was suppressed to 15%. This evidence proved that as training progressed, our algorithm's winning rate against the expert-level bot improved significantly. At the end of the training, the MAHPG was able to defeat the bot easily.

4.4. Ablation study of hierarchical hybrid policy networks

An ablation study was conducted to demonstrate the effectiveness of the proposed hierarchical policy architecture. After 50 iterations of agent training, ten random air combat episodes were simulated. Under such an experiment setting, two typical air combat maneuvers, guided level flight and beaming, were adopted as high-level action outputs in both plain policy and hierarchical policy architectures. The probability distribution of low-level actions can then be accounted for. On the contrary, the plain policy treats high-level and low-level actions independently.

As shown in Fig. 7, under the guided level flight maneuver, the n_{nc} and v_c action distributions output by plain policy show larger variance, while the hierarchical policy distribution scatters with significantly smaller variance. Moreover, the lower-level action distribution under hierarchical policy performs much more closer to the steering trajectory of human experts (Bonanni, 1993). This phenomenon can also be revealed in the steering command distribution under the beaming maneuver. It demonstrates that the additional high-level action information, which is passed to the low level by the proposed hierarchical policy architecture, reduces the uncertainty of the low-level actions to a certain extent. Moreover, the learned low-level actions perform more suitable for the specified high-level maneuver. By comprehensively analyzing the battle results, it was found that the proposed mechanism effectively improves the overall coordination of the algorithm hierarchy and achieves a higher winning rate.

5. Discussion and analysis of emergent behaviors

This section is mainly focused on the air combat tactics emergence process and the automatic knowledge innovation mechanism behind them (refer to Appendix E for tactics terms description). First, the discovered ATIA phenomenon will be introduced. Then, three categories of emergent behaviors will be described in detail. Finally, the sensitivity of batch size selection for tactics emergence will be investigated.

5.1. Discovery of ATIA phenomenon

Innovation may be explained by considering that the environment to which units adapt can change over time. This causes old adaptations to lose their applicability and thereby motivates exploration toward new innovative solutions, which is equivalent to learning by progressively adapting to the underlying changing environment (Joel et al., 2019). Meaningful skills and behaviors may emerge naturally from the non-stationary dynamics of multi-agent interaction processes, without any need for environmental engineering. This facilitates the acquisition of complex behaviors that would not be learned otherwise (Narvekar,

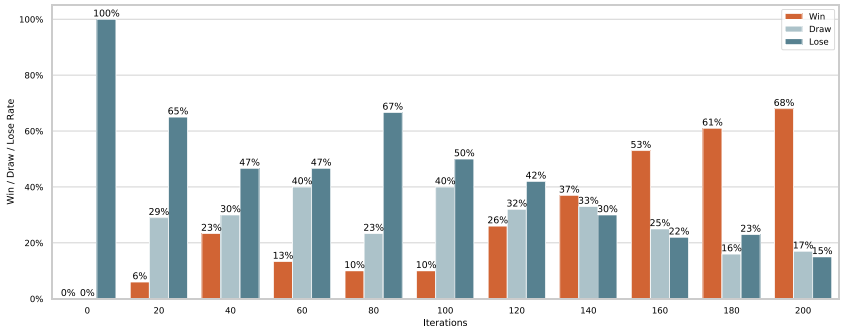
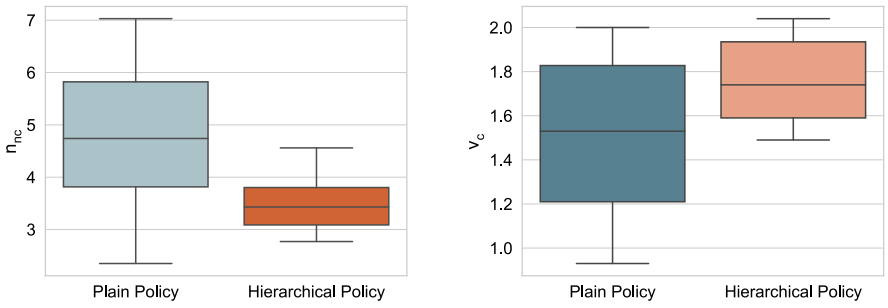
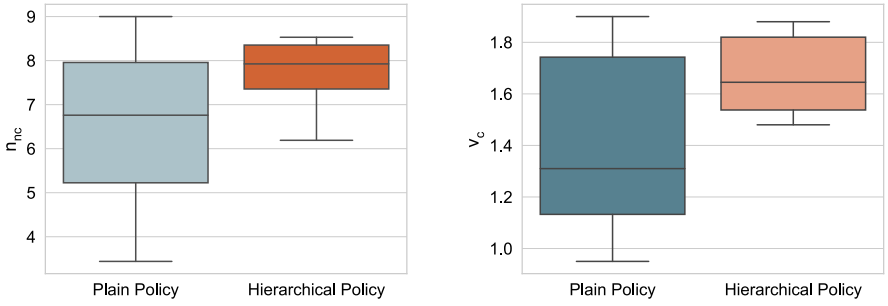


Fig. 6. Competition result with expert level hand-crafted bot.



(a) n_{nc} comparison for guided level flight (b) v_c comparison for guided level flight



(c) n_{nc} comparison for beaming (d) v_c comparison for beaming

Fig. 7. Comparison of steering commands learned with/without policy hierarchy.

2017; Czarnecki et al., 2018). Similarly, throughout the history of modern air combat, new tactics innovated by humans came from accurate cognition and adaption with the environment, especially from the evolution of the opponent's behavior. Once new behaviors have been adapted, this emerged strategy will provide new competitive pressure for corresponding opponents. This is consistent with the emergence mechanism of the multi-agent interaction process as aforementioned: both sides' confrontation is expected to continue developing new air combat tactical behaviors by continually adapting to the changes of opponents.

With the progress of continuous MARL air combat agent training in the WUKONG simulation environment (refer to Section 4.1), the adversarial agents gradually produce a series of brand-new tactical behaviors. Although these behaviors were not hard-coded by human experts, they are highly consistent with expert-level jet pilot performance (Bonanni, 1993; Shaw, 1985). It is worth noting that the agent learned from nothing, only using episodic rewards and sparse key air combat event based shaped rewards. Therefore, adversarial agents constructed an adaptive course learning mechanism through multi-agent gaming. This phenomenon we discovered, i.e., gradually creating brand new tactical behaviors as air combat training progresses and forcing opponents to adapt accordingly, is named Air Combat Tactics Interplay Adaption (ATIA). In a sense, the discovery of ATIA indicates the first time that we have endowed air combat AI with the ability of continuous tactics evolution.

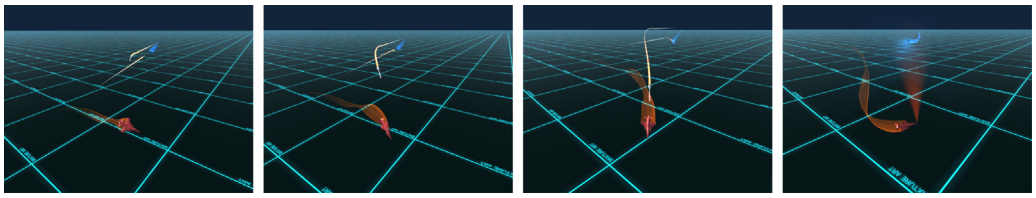
At the very beginning, agents spun around without showing any meaningful tactical behavior. As agents gradually enhanced their aiming accuracy and shooting skills through self-play, we noticed that adversarial agents began to shoot each other down. Once key event occurred, e.g., shooting down, the opponent was able to evade incoming missiles by exploring and mastering defense dragging tactics. This interaction between agents along with the training process effectively promoted the continuous progression of adversarial evolution. Experiments also indicate that meaningful behaviors only emerge based on the KAERS mechanism with batch size 32 000 (refer to Section 5.5). In summary, the spontaneously emerged air combat tactics with training are processed into the following three categories.

5.2. Dive and retreat tactics

After 75 training iterations, the earliest useful tactical behavior category discovered by agents is called dive and retreat, which can be divided into three concrete tactics, as shown in Table 4. The trajectory of this category of behavior can be seen in Fig. 8. When the blue aircraft launched a mid-range missile, the red aircraft consciously carried out a defensive turn-out and retreat tactic. With up to around 120 training iterations, the red one learned to stop tactical dragging immediately before the missile was about to hit, and thus, created more offensive opportunities, which were named precise turn-out retreat timing tactics. When training lasted about 230 iterations, the red one developed a novel maneuvering strategy, i.e., when the blue one launched a missile at a closer range, it decisively carried out a turn-out and dive maneuver, thereby completing the effective evasion of high-threat missiles when only horizontal turning-out maneuvers would not work. Given that atmospheric density increases exponentially with altitude decrease, missiles out of rocket propulsion will decelerate quickly. On the contrary, the thrust of the red aircraft will increase rapidly as the altitude decreases, thereby significantly increasing the energetic advantage from the pursuit missile. Consequently, the emergence of these behaviors demonstrates that the agents have learned to use mathematical laws of atmospheric density to develop meaningful tactics.

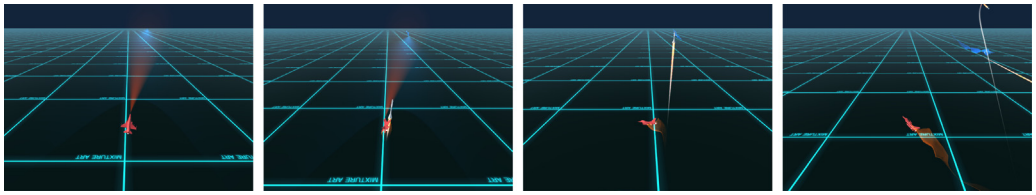
5.3. Loft shoot and crank tactics

The next emerged behavior category was loft shoot and crank tactics, which also include the three concrete tactics listed in Table 4 (see Fig. 9). When training began, both shooting behaviors between adversarial sides were chaotic and irregular. After approximately 110 iterations, the red aircraft gradually learned to accurately grasp missile launch timing by a precise shot timing tactic. From the empirical study, the shooting behavior of both sides then began to show increasing lethality and regularity. It is worth noting that when we canceled the



(a) Shooting and approaching (b) Defensive dragging after shot (c) Diving (d) Re-engaging

Fig. 8. Dive and retreat tactics emergence.



(a) Opponent approaching and target locking (b) Climbing (c) Cranking (d) Missile guiding

Fig. 9. Loft shoot and crank tactics emergence.

shot key event shaped reward defined in Table 1, the shooting remained irregular. After about 270 iterations, when the attacking distance was too great for firing, the red aircraft even learned to increase the overall energy of the aircraft by a climb accelerate and shoot tactic, thereby enlarging the effective range of the missiles. At about the 380th iteration, the red aircraft surprisingly created a sequence of maneuvers to reduce the closing rate of the opponent's incoming missiles by performing a post-launch Single Side Offset (SSO) maneuver, defined in Table 6. This further improved its probability of survival, and we name this mannerism the shoot and crank tactic. Interestingly, the spontaneous behaviors learned through ATIA are completely consistent with expert BVR air combat jet pilots as described in Bonanni (1993).

5.4. Drag and re-engage tactics

The final emerged behaviors category was quite complicated and began to show a strong sense of confrontation and offensiveness, namely drag and re-engage tactics, as shown in Fig. 10. After approximately 200 iterations, the agent first learned to utilize the beaming maneuver to perform a precise drag timing tactic. When training reached around 430 iterations, the red aircraft even deprecated previously learned conservative missile escaping tactics and instead flew laterally to appropriately consume the incoming missile's energy, thereby ensuring that it could dynamically switch between offensive and defensive strategies at any time. When the incoming missile seemed nearly ineffective, the agent could accurately use the opportunity to re-engage and instantly switched to offense, completing the advanced tactical behavior of U-turn shooting when it turned close to the aiming LOS. The emergence of this kind of behavior indicates that agents could fully grasp the quantitative relationship of air combat situations according to the opponent's behavior. After that, self-play training maintained a relatively high level of adversarial confrontation benchmark and continued to produce complex and mutable flight trajectories, which are not yet recognized by humans.

5.5. Batch size sensitivity

As shown in Table 4, we found that the training batch size adopted in each iteration had a critical impact on ATIA behavior emergence. When batch size reduced to 2000, agents could not learn any useful tactical behaviors by simply increasing the training iterations. When the batch size reached 16000, agents could only learn basic shooting and turning behaviors through self-play. However, from empirical studies, learned behaviors always emerged with uncertainty that could not be ignored, rather than converging to specific behaviors for particular situations. Finally, when the batch size was 32000, agents could spontaneously emerge all the tactical behaviors. Due to computational resource limitations, we did not conduct larger-scale distributed sampling-training iterative experiments. Please refer to Appendix A for more details.

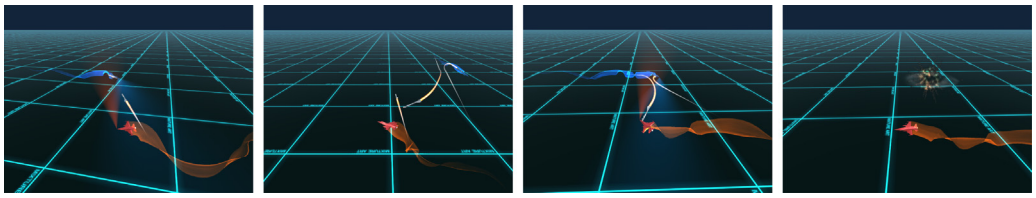
Table 4
Progressive emergence of air combat tactics.

Emergent tactics category	Concrete-tactics	Emergence batch/iteration		
		2000	16 000	32 000
Dive and retreat tactics	Turn-out retreat	–	210	75
	Precise turn-out retreat timing	–	360	120
	Turn-out and dive	–	–	230
Loft shoot and crank tactics	Precise shoot timing	–	290	110
	Climb, Accelerate and shoot	–	–	270
	Shoot and crank	–	–	380
Drag and re-engage tactics	Precise drag timing	–	–	200
	Precise re-engage timing	–	–	430

Actually, in common sense, one-on-one BVR air combat is a fairly complex multi-agent competition problem. As the training batch size increases, the estimation of the sample-based objective function through air combat training becomes accurate. The neural network policy gradient's variance could effectively reduce as the batch samples increase, thereby preventing the learning process from falling into degradation induced by gradient estimation variances.

5.6. Further discussion

Generally speaking, air combat scenarios can be divided into two categories: beyond-visual-range (BVR) and within-visual-range (WVR). Since BVR scenarios have come to make up the majority of modern air-to-air engagements (Stillion, 2015), it is more practical to conduct this AI research with BVR settings.



(a) Defensive dragging against incoming missile (b) Keep on dragging (c) Precise re-engage maneuvering (d) Survived

Fig. 10. Drag re-engage and shoot tactics emergence.

More specifically, in the self-play training process, the higher the training reward is, the stronger the agent's ability will be. But as in BVR circumstances, the only principle is to gain air dominance by eliminating all the enemies, which has no strong causal relationship to the expert hand-crafted training rewards. Pursuing such kind of rewards is proven to influence the confrontation performance negatively by facing the model over-fitting problem. This is the reason that the A2C with McGraw score algorithm performed mediocrely in terms of winning rate even though it achieves the highest reward as discussed in Section 4.2.

To address this issue, we adjust the weight settings of the proposed KAERS mechanism by these principles. Firstly, the episodic rewards have direct causal relations to the winning rate, and thus, should be enlarged significantly against the event-based rewards. Furthermore, the event-based rewards offer denser returns which helps the agents to learn basic air combat skills in the early training stage, such weights are adjusted much smaller than the episodic rewards. This unique design achieves a balance between the objective mission goal consideration and the skill learning efficiency as demonstrated in Section 4.2.

In the field of deep reinforcement learning, scenarios containing only two agents, e.g., agents from two opposite sides, respectively, can be considered as fully-competitive multi-agent scenarios (Hernandez-Leal et al., 2019; Berner et al., 2019; Vinyals et al., 2019). Besides, if the setting involves more than two agents, such a scenario is more in line with the public's cognition of multi-agent. However, more challenges will be faced if more agents are involved in air combat, e.g., consensus, cooperation, and communication. Addressing such challenges will be our main research direction in the future.

6. Conclusion

In this paper, a MAHPG algorithm was proposed to address the hierarchical discrete and continuous hybrid decision-making problem in air combat. It can learn operational strategies only by self-play. Moreover, the ATIA phenomenon was discovered during training. It inspired battle aircraft to create a step-by-step series of air combat curriculum by progressively adapting the mutated strategy of the opponent coherently. In a sense, the MAHPG gained continuous tactics evolutionary capacity. In the experiment, a comparison with four state-of-the-art air combat methods and an expert-level bot revealed that the proposed MAHPG demonstrated noticeable superiority in terms of defense and offense. In the future, the MAHPG can be applied to practical applications with complex and hybrid decision-making problems, such as unmanned vehicles, autonomous robots, financial risk control, and infectious disease prevention and control. Besides, we will work toward a POMDP version of the one-on-one BVR air combat simulation and try to investigate solutions for the high uncertainty nature of BVR air combat.

CRediT authorship contribution statement

Zhixiao Sun: Methodology. **Haiyin Piao:** Conceptualization, Methodology, Data curation, Writing - original draft, Visualization, Software. **Zhen Yang:** Software. **Yiyang Zhao:** Formal analysis. **Guang Zhan:** Methodology. **Deyun Zhou:** Resources, Investigation. **Guanglei Meng:** Validation. **Hechang Chen:** Writing - review & editing. **Xing Chen:** Writing - review & editing. **Bohao Qu:** Writing - review & editing. **Yuanjie Lu:** Writing - review & editing.

Declaration of competing interest

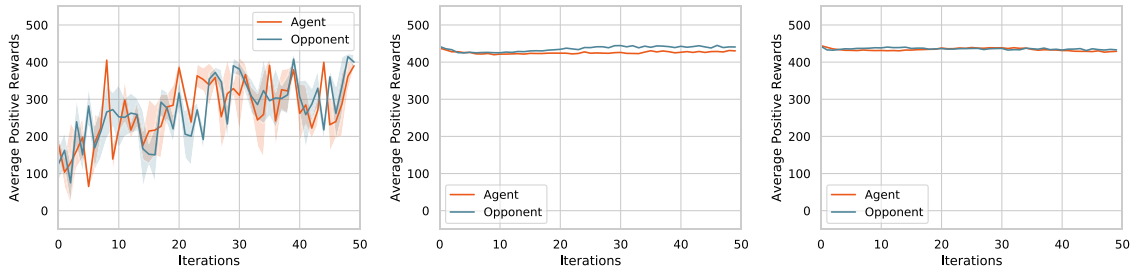
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Batch size effectiveness analysis

Fig. 11(a)–(c) compares the performance of the proposed MAHPG algorithm on different iteration batch sizes. The shaded region denotes a standard deviation of average evaluations over five trials. First, it can be clearly seen that the average rewards gradually increased from a high negative score, which means that our algorithm can drive an agent to explore the air combat environment and gradually learn meaningful tactical behavior through self-play from scratch, rather than hand-crafted from expert knowledge or dense reward signals (McGrew et al., 2010). We also observed that the adversarial agents' scores gained gradually converged to approximately 100.0 to −50.0 during the entire training process and no longer increases. This is mainly because an agent will lose 25 points once the shooting event occurs, and the score convergence trend shows that our algorithm could gradually approach a superior fixed point through training. Second, we found that all three different training batch sizes can drive the agents' average rewards to continue rising as training progresses. According to the visual representation, both the agent and opponent exhibited continuous learning behavior from a chaotic and meaningless flight from the very beginning to start approaching with accurately fired missiles. The training process with a batch size of 16 000 and 32 000 had allowed agents to rise to an average reward of approximately −100.0 and stabilized in the first ten iterations. However, when the batch size was reduced to 2000, although adversarial agents can reach −100.0 points through self-play, this specific process lasted nearly 50 iterations, which was performed much slower than the large batches. Therefore, we realize that a larger batch size can significantly improve the convergence of self-play for this air combat problem. Since a successful kill gains 500 points immediately (considering zero-sum games, the corresponding killed player will lose 500 points), Fig. 12(a)–(c) focuses on calculating the average-positive rewards. We could also see that when the batch size is too small, it is difficult for the algorithm to learn shooting and killing behaviors, which means that the average positive reward was accompanied by huge noise as the training proceeded. Consequently, larger batch sizes result in more stable and repeating shooting behaviors, therefore providing a more meaningful learning signal.

(a) Batch size = 2000 (b) Batch size = 16000 (c) Batch Size = 32000

Fig. 11. Average self-play training rewards.



(a) Batch size = 2000 (b) Batch size = 16000 (c) Batch Size = 32000

Fig. 12. Average positive self-play training rewards.

Appendix B. Proof of value function bias existence under POMDP

The value of a given policy π of a POMDP can be represented in the following matrix form below, in which O is the observation function specifies the conditional observation probabilities:

$$V^{\pi} = R + \gamma PO\pi V \tag{18}$$

Once we use the approximate models (i.e., neural networks) instead of the ground truth models to calculate the value of a given policy π, we will have:

$$\hat{V}^{\pi} = \left(I - \gamma \hat{P}\hat{O}\pi\right)^{-1} R \tag{19}$$

where $\hat{V}^{\pi}$, $\hat{P}$, and $\hat{O}$ are corresponding approximations. Since observation and system intrinsic noise always exists in a practical system, $\hat{P}$ and $\hat{O}$ likely to be biased with error terms $\tilde{P}$ and $\tilde{O}$, we assume these error terms are independent and zero biased.

$$\hat{P} = P + \tilde{P}, \quad \hat{O} = O + \tilde{O} \tag{20}$$

$$\hat{V} = (I - \gamma(P + \tilde{P})(O + \tilde{O})\pi)^{-1} R \tag{21}$$

Applying matrix form Taylor expansion, the above equation can be transformed into:

$$\hat{V} = \sum_{k=0}^{\infty} \gamma^k f_k(\tilde{P}, \tilde{O}) R \tag{22}$$

In which:

$$X = (I - \gamma PO\pi)^{-1} \tag{23}$$

$$f_k(\tilde{P}, \tilde{O}) = (X(\tilde{P}O\pi + P\tilde{O}\pi + \tilde{P}\tilde{O}\pi))^k X \tag{24}$$

Thus the expectation of $\hat{V}^{\pi}$ can be re-written in such form:

$$\mathbb{E}[\hat{V}] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k f_k(\tilde{P}, \tilde{O}) R\right] \tag{25}$$

Ignoring higher order derivatives we obtain a second order approximation:

$$\mathbb{E}[\hat{V}] = X R + \gamma \mathbb{E}\left[f_1\right] R + \gamma^2 \mathbb{E}\left[f_2\right] R \tag{26}$$

Since $\tilde{P}$ and $\tilde{O}$ are zero mean, the middle term will be eliminated. After this simplification, the POMDP value function is expected to have an non-zero bias term rather than approximately zero-biased in MDP form.

$$\mathbb{E}[\hat{V}] = V + \gamma^2 \mathbb{E}\left[f_2(\tilde{P}, \tilde{O})\right] R \tag{27}$$

Table 5
Air combat states definition.

State variable	Meaning
x	Agent aircraft position in global axis
x_b	Opponent aircraft position in global axis
v	Agent aircraft velocity in global axis
v_b	Opponent aircraft velocity in global axis
ψ	Yaw angle
θ	Pitch angle
ϕ	Roll angle
n_n	Normal load factor
r	Relative range between agent and opponent
$\dot{r}$	Closing rate of relative range between agent and opponent
AA	Aspect angle based on LOS
ATA	Antenna train angle based on LOS
EL	Opponent elevation angle based on horizontal plane
Δh	Opponent Δh
lo	Boolean signal indicates if airborne radar have locked on bandit
$warn$	Locked signal by opponent's radar or incoming missile
m_{left}	Missiles left in weapon bay
T_{go}	Time left till mid-range missile's seeker go active

Appendix C. Air combat states definition

The BVR air combat states are described in Table 5:

Appendix D. Air combat actions definition

The BVR air combat actions are described in Table 6:

Table 6
Basic Fighter Maneuvering (BFM) macro actions.

Categories	Serial number	Macro actions	n_{nc}	v_c (mach)
Offensive	1	Guided level flight	2.0	0.85
	2	+30° Climbing and accelerating	3.0	1.3
	3	+60° Climbing and accelerating	3.0	2.0
	4	−30° Offensive descending	4.0	0.9
	5	−60° Offensive descending	4.0	0.9
	6	±30° Single Side Off(SSO)[a]	3.0	0.9
	7	±50° Single Side Off(SSO)	3.0	0.9
	8	Horizontal snake maneuvering	3.0	0.9
Defensive	9	Split-S	8.0	2.0
	10	Beaming	3.0	0.9
Retreat	11	Level turning	3.0	0.9
	12	Fast turning	5.0	2.0
	13	Descending −30° after turning	5.5	2.0
	14	Descending −60° after turning	7.0	2.0

[a]The ± symbol denotes that aircraft will automatically maneuvering to the smaller angle offset direction.

Appendix E. Air combat tactics terms

The air combat tactics terms mentioned in this article are explained as follows:

- Defensive dragging: Horizontally turn 90° for breaking away from enemy missiles.
- Diving: Continuously lower down the altitude and turn 180° to retreat.
- Re-engaging: Turn back to attack at an appropriate time.
- Target locking: Lock the opponent by radar.
- Climbing: Gain more altitude by setting up an upgoing course.
- Cranking: Horizontally turn 50° and keep the enemy in track in the radar system, in order to reduce the closing rate to the opponent.
- Missile guiding: Lock the opponent by radar and continuously transmit its location and velocity information to the on-the-fly missile.
- Loft shoot: Keep the aircraft nose up and shoot the missile with some upward angle to gain longer missile attack range.
- Beaming: Make the course perpendicular to the opponent's course.
- Single Side Off (SSO): Make the heading azimuth angle off the line of sight with a certain degree.
- Split-S: Half-rolls the aircraft inverted and executes a descending half-loop, resulting in level flight in the opponent's direction at a lower altitude.

References

Altan, A., Aslan, O., Hacıoglu, R., 2018. Real-time control based on NARX neural network of hexarotor UAV with load transporting system for path tracking. In: 2018 6th International Conference on Control Engineering Information Technology (CEIT), Istanbul, Turkey, pp. 1–6.

Altan, A., Hacıoğlu, R., 2020. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. Mech. Syst. Signal Process. 138, 106548.

Altan, A., Karasu, S., 2019. The effect of kernel values in support vector machine to forecasting performance of financial time series and cognitive decision making. J. Cogn. Syst. 4, 17–21.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dkebiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al., 2019. Dota 2 with large scale deep reinforcement learning. https://cdn.openai.com/dota-2.pdf/(accessed 13 August 2020).

Bonanni, P., 1993. The Art of the Kill. Spectrum HoloByte, California.

Burgin, G.H., 1976. Improvements to the Adaptive Maneuvering Logic Program. Tech. Rep., NASA Contractor Report.

Burgin, G.H., Eggleston, D.M., 1976. Design of an all-attitude flight control system to execute commanded bank angles and angles of attack. Tech. Rep., NASA Contractor Report.

Burgin, G.H., Fogel, L.J., Phelps, J.P., 1975. An Adaptive Maneuvering Logic Computer Program for the Simulation of One-On-One Air-To-Air Combat. Tech. Rep., NASA Contractor Report.

Byrnes, M., 2014. Nightfall: Machine autonomy in air-to-air combat. Air Space Power J. 28, 48–75.

Campbell, M., Hoane Jr, A.J., Hsu, F.-h., 2002. Deep blue. Artif. Intell. 134 (1–2), 57–83.

Czarnecki, W.M., Jayakumar, S.M., Jaderberg, M., Hasenclever, L., Teh, Y.W., Heess, N., Osindero, S., Pascanu, R., 2018 .Mix & Match Agent Curricula for Reinforcement Learning. In: Proceedings of the 35th International Conference on Machine Learning, Vol. 80, Stockholm, Sweden, pp. 1095–1103.

Ding, N., Soricut, R., 2017. Cold-start reinforcement learning with softmax policy gradient. In: Advances in Neural Information Processing Systems, Long Beach, Los Angeles, USA, pp. 2817–2826.

Ernest, N., Carroll, D., Schumacher, C., Clark, M., Cohen, K., Lee, G., 2016. Genetic fuzzy based artificial intelligence for unmanned combat aerial vehicle control in simulated air combat missions. J. Def. Manag. 6 (1), 2167–0374.

Floyd, M.W., Karneeb, J., Moore, P., Aha, D.W., 2017. A goal reasoning agent for controlling UAVs in beyond-visual-range air combat. In: Sierra, C. (Ed.), Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org, pp. 4714–4721.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, pp. 249–256.

Goodrich, K.H., 1993. A hIgh-Fidelity, Six-Degree-Of-Freedom Batch Simulation Environment for Tactical Guidance Research and Evaluation, Vol. 4440. NASA, Scientific and Technical Information Program, Washington.

Goodrich, K., McManus, J., 1989. Development of a tactical guidance research and evaluation system (TGRES). In: Flight Simulation Technologies Conference and Exhibit, Boston, MA, U.S.A, p. 3312.

Heinrich, J., Lanctot, M., Silver, D., 2015. Fictitious self-play in extensive-form games. In: International Conference on Machine Learning, Lille, France, pp. 805–813.

Hernandez-Leal, P., Kartal, B., Taylor, M.E., 2019. A survey and critique of multiagent deep reinforcement learning. Auton. Agents Multi-Agent Syst. 33 (6), 750–797.

Jaderberg, M., Czarnecki, W.M., Dunning, I., Marris, L., Lever, G., Castaneda, A.G., Beattie, C., Rabinowitz, N.C., Morcos, A.S., Ruderman, A., et al., 2018. Human-level performance in 3D multiplayer games with population-based deep reinforcement learning. Science 364, 859–865.

Joel, L., Edward, H., Marc, L., Thore, G., 2019. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. arXiv preprint arXiv:1903.00742.

Kurniawan, B., Vamplew, P., Papasimeon, M., Dazeley, R., Foale, C., 2019. An empirical study of reward structures for actor-critic reinforcement learning in air combat manoeuvring simulation. In: Australasian Joint Conference on Artificial Intelligence. Springer, Adelaide, Australia, pp. 54–65.

Leslie, D.S., Collins, E.J., 2006. Generalised weakened fictitious play. Games Econom. Behav. 56 (2), 285–298.

Littman, M.L., 1994. Markov Games as a framework for multi-agent reinforcement learning. In: Cohen, W.W., Hirsh, H. (Eds.), Machine Learning Proceedings 1994. Morgan Kaufmann, San Francisco, CA, USA, pp. 157–163.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, Long Beach, Los Angeles, USA pp. 6379–6390.

McGrew, J.S., How, J.P., Williams, B., Roy, N., 2010. Air-combat strategy using approximate dynamic programming. J. Guid. Control Dyn. 33 (5), 1641–1654.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, New York, USA, pp. 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529.

Narvekar, S., 2017. Curriculum learning in reinforcement learning. In: IJCAI, Melbourne, Australia, pp. 5195–5196.

Osinga, F.P., 2007. Science, Strategy and War: The Strategic Theory of John Boyd. Routledge, London.

Ramirez, M., Papasimeon, M., Lipovetzky, N., Benke, L., Miller, T., Pearce, A.R., Scala, E., Zamani, M., 2018. Integrated hybrid planning and programmed control for real time UAV maneuvering. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, Stockholm, Sweden, pp. 1318–1326.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Shaw, R.L., 1985. Fighter Combat. Naval Institute Press Annapolis, Annapolis.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., 2016. Mastering the game of go with deep neural networks and tree search. Nature 529 (7587), 484.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al., 2017. Mastering the game of go without human knowledge. Nature 550 (7676), 354–359.

Stillion, J., 2015. Trends in Air-To-Air Combat: Implications for Future Air Superiority. Tech. Rep., Center for Strategic and Budgetary Assessments.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT press, Cambridge.

Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al., 2019. Grandmaster level in starcraft II using multi-agent reinforcement learning. Nature 575 (7782), 350–354.

Virtanen, K., Karelahti, J., Raivio, T., 2006. Modeling air combat by a moving horizon influence diagram game. J. Guid. Control Dyn. 29 (5), 1080–1091.

Virtanen, K., Raivio, T., Hämäläinen, R.P., 2002. An influence diagram approach to one-on-one air combat. In: Proceedings of the 10th International Symposium on Differential Games and Applications, Vol. 2. St. Petersburg, Russia, pp. 859–864.

Xiong, J., Wang, Q., Yang, Z., Sun, P., Han, L., Zheng, Y., Fu, H., Zhang, T., Liu, J., Liu, H., 2018. Parametrized deep Q-networks learning: Reinforcement learning with discrete-continuous hybrid action space. arXiv preprint arXiv:1810.06394.