electronics

MDPI

Article

Research on Efficient Multiagent Reinforcement Learning for Multiple UAVs' Distributed Jamming Strategy

Weizhi Ran [1], Rong Luo [2], Funing Zhang [3], Renwei Luo [1] and Yang Xu [1,*]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; imrwz0710@std.uestc.edu.cn (W.R.)
[2] Naval Research Academy, PLA, Beijing 100161, China; luorong583@163.com
[3] School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; funingz@xauat.edu.cn
* Correspondence: xuyang@uestc.edu.cn

Abstract: To support Unmanned Aerial Vehicle (UAV) joint electromagnetic countermeasure decisions in real time, coordinating multiple UAVs for efficiently jamming distributed hostile radar stations requires complex and highly flexible strategies. However, with the nature of the high complexity dimension and partial observation of the electromagnetic battleground, no such strategy can be generated by pre-coded software or decided by a human commander. In this paper, an initial effort is made to integrate multiagent reinforcement learning, which has been proven to be effective in game strategy generation, into the distributed airborne electromagnetic countermeasures domain. The key idea is to design a training simulator which close to a real electromagnetic countermeasure strategy game, so that we can easily collect huge valuable training data other than in the real battle ground which is sparse and far less than sufficient. In addition, this simulator is able to simulate all the necessary decision factors for multiple UAV coordination, so that multiagents can freely search for their optimal joint strategies with our improved Independent Proximal Policy Optimization (IPPO) learning algorithm which suits the game well. In the last part, a typical domain scenario is built to test, and the use case and experiment results manifest that the design is efficient in coordinating a group of UAVs equipped with lightweight jamming devices. Their coordination strategies are not only capable of handling given jamming tasks for the dynamic jamming of hostile radar stations but also beat expectations. The reinforcement learning algorithm can do some heuristic searches to help the group find the tactical vulnerabilities of the enemies and improve the multiple UAVs' jamming performance.

Keywords: multiagent reinforcement learning; IPPO learning algorithm; multiple UAVs distributed jamming strategy

Citation: Ran, W.; Luo, R.; Zhang, F.; Luo, R.; Xu, Y. Research on Efficient Multiagent Reinforcement Learning for Multiple UAVs' Distributed Jamming Strategy. *Electronics* **2023**, *12*, 3874. https://doi.org/10.3390/electronics12183874

Academic Editor: Mehdi Sookhak

Received: 17 August 2023
Revised: 7 September 2023
Accepted: 12 September 2023
Published: 14 September 2023

1. Introduction

With the development of multiagent reinforcement learning and UAV technologies, research on efficient multiagent reinforcement learning for coordinating multiple UAV distributed jamming strategies has gradually become an active area of study. The "unmanned" characteristics of UAVs enable them to perform countermeasure operations that would be infeasible or too dangerous for human pilots. However, due to the limited payload capacity, lightweight equipment, and low power of individual UAVs, it is difficult for a single UAV to independently complete an entire operational mission [1]. As a result, coordinated operations with multiple UAVs have become a key development direction for intelligent warfare systems of the future. Multiple UAV coordination can fully harness the capabilities of an entire swarm, enabling superior coordination and intelligence. By distributing control across multiple low-cost, rapidly reconfigurable UAVs with lightweight payloads, this approach can significantly improve coordination efficiency and task performance [2,3].

Through distributed control, groups of UAVs can overcome the limitations of individual UAVs' local perceptions and capabilities, playing an indispensable role across application domains [4]. This paper focuses on the coordinated control problem in the context of radar jamming scenarios [5]. Constructing realistic electromagnetic countermeasure scenarios and fielding physical UAV teams involves tremendous expense. Additionally, directly testing insufficiently validated multi-UAV control models in live environments risks irrecoverable losses if failures occur. Consequently, existing research methods often involve first building simulation environments to iterate on and verify control approaches before final live testing [6,7]. By developing a task-tailored simulator and iteratively refining UAV coordination models within it, this approach greatly reduces costs and increases iteration efficiency prior to real-world deployment [8].

In recent years, reinforcement learning has become a highly active research domain given its ability to learn complex tasks with minimal reliance on prior sample data or explicit programming. Reinforcement learning has achieved major successes across gaming domains, including mastering Go [9,10], Starcraft [11], and Dota 2 [12]. Inspired by these proven capabilities, we make early efforts to integrate multiagent reinforcement learning into distributed electromagnetic countermeasures [13,14]. The complexity and need for highly flexible strategies in coordinating multiple UAVs for efficient distributed radar jamming naturally suit reinforcement learning. The scarcity of historical countermeasure data makes it difficult to derive optimal policies from prior human experience. For problems lacking sufficient training samples, reinforcement learning offers clear advantages. It can optimize policies through continuous exploration without dependence on large existing datasets.

Reinforcement learning has shown immense promise in training artificial agents to excel in complex, high-dimensional environments. A reinforcement learning agent learns by interacting with its environment, receiving feedback in the form of rewards or penalties based on its actions. This enables the agent to incrementally improve its decision policy to maximize cumulative reward. Deep reinforcement learning combines reinforcement learning principles with deep neural networks, leveraging deep learning's representation power to tackle problems with high-dimensional state and action spaces. This approach has achieved remarkable results across challenging domains including video games [11,12], robotics [15], etc.

For multi-UAV coordination, traditional approaches such as vector fields have been widely used to achieve UAV team coordination and formation control [16,17]. Despite, its strengths in coordinating small dynamic UAV teams, it has not shown vivid research results in coordinating larger members of UAVs in complex environments, such as electronic warfare. However, deep reinforcement learning offers key strengths. It allows agents to learn coordinated policies based on experiential simulated training rather than human design. Agents can explore possible action spaces and learn optimal joint policies exceeding human performance [9,10]. Compared to supervised or unsupervised deep learning, reinforcement learning provides an interactive learning loop enabling agents to continually refine behavior through environmental feedback. While deep learning alone requires large labeled datasets which can be scarce in novel domains like radar jamming, reinforcement learning can compensate through online simulation.

This simulation-based deep reinforcement learning approach provides multiple benefits. It allows extensive iteration on agent policies, hyperparameters, and environmental parameters which may be infeasible with physical UAVs. Simulation facilitates the generation of abundant, task-relevant training data. It provides interpretability into agent behavior and an interactive debugging environment. Policies can be empirically validated against simulated scenarios before real-world deployment. This approach combines the strengths of deep reinforcement learning and simulation to tackle multi-UAV coordination for radar jamming.

Thus, this paper focuses on the iteration and optimization of reinforcement learning models applied to the generation of multiple UAV distributed jamming strategies. The key

idea is to design a training simulator that is close to a real electromagnetic countermeasure strategy game, and then generate effective strategies. Based on the current application of multiagent in this field, the paper first designs a model for multiagent reinforcement learning and defines a paradigm for it. Then, based on the existing PPO model learning algorithm, a new learning algorithm, the IPPO learning algorithm is proposed. After that, the electromagnetic countermeasure scenario of multiple UAVs is modeled and deduced in the simulator, and the results are analyzed by training curves. The experimental results illustrated that this strategy is highly effective for multiple UAV jamming tasks with tactical vulnerability analysis.

2. Multiagent Reinforcement Learning Model

Reinforcement learning [18] aims to make the agent learn the appropriate behavior in a specific state, through the reward signal obtained by interacting with the environment. Therefore, reinforcement learning algorithms do not need to explicitly state how a task should be accomplished. Instead, under the guidance of trying various actions and reward information, the agent gradually obtains a good decision method in a trial-and-error manner. Reinforcement learning algorithms usually have good generality, and they can be competent for many different types of tasks without exploring specific mysteries.

If the environment can be fully described by the current state, it can be called the Markov Decision Process. For practical reasons, reinforcement learning is usually modeled by the Markov process. Because the agent needs to consider long-term rewards, it must be able to learn from delayed reward signals. In other words, the agent may go through a long decision-making sequence, eventually reaching a state of high reward. The state is usually the completion of the specified task. The agent measures the quality of a strategy, which needs to be determined by an evaluation index of cumulative rewards. Reinforcement learning typically uses an objective function of the form to estimate the expected cumulative reward of a strategy.

$$\eta(\pi) = E_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t r(S_t)] \tag{1}$$

τ is a sequence of states and actions. $r(S_t)$ is a reward in the current state. $E_{\tau\sim\pi}$ is expectation under the current strategy. π is a stochastic strategy. γ is called the discount factor. Typically, the value of γ should be less than 1. On the one hand, it is to ensure the convergence of the formula; on the other hand, the future benefits have not been obtained. Therefore, the more distant the expected reward, the more underestimated it is.

The control problem of multiple UAVs belongs to multiagent reinforcement learning [19,20]. Each UAV is an agent and needs to be defined according to the quintuple. The structure of its interaction with the environment is shown in Figure 1:

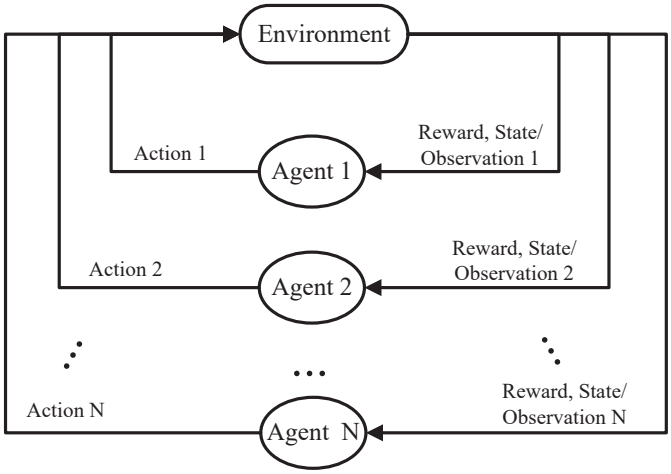

Figure 1. Multiagent reinforcement learning interaction paradigm.

The environment, action, state, reward, and discount factors in the above model comprise the quintuple of reinforcement learning. The quintuple is represented below:

$$(S, A, P, r, \gamma) \tag{2}$$

S represents the state set of environment, A represents the action-set of the agent, $P : S \times A \times S \to R$ represents the state transition probability of the environment, $r : S \to R$ represents the specified reward function, and γ represents the discount factor of delayed reward. The Markov Decision Process (MDP) of reinforcement learning can be fully characterized by this quintuple, subsequent theories and reviews are based on the model.

3. Multiple UAVs Distributed Jamming Strategy

In reinforcement learning, a neural network is usually constructed to represent the value function or strategy of the agent [21,22]. According to the current state, the method of learning the value function can output the expected utility Q of each action. The largest value is the optimal action. According to the current state, the method of learning strategy [23–25] is to output the execution probability of each action under the optimal strategy. The optimal action can be obtained by sampling under the probability.

The Proximal Policy Optimization (PPO) algorithm [26] is a method with stable training and good generalization in current reinforcement learning. In single-agent strategy generation, the model optimization process is as follows:

As shown in Figure 2, based on these steps, a corresponding strategy output is obtained:

1. First, two neural network models are defined, which can be initially random parameters, Actor and Critic. The Actor is called an action network, whose input is the state of the agent and the output is the probability of all executable actions of the agent. The Critic is called the evaluation network, whose input is the state of the agent and the output is the state value. The size of the value indicates the quality of the state, which is used to guide the action network for better learning. The network represents the strategy of the agent. Our goal is to learn such a strategy that maximizes the long-term reward (expected reward) of the agent in the environment.
2. Load the current strategy into the simulator and interact with the environment for a certain number of times, such as 1000 or more. Each interaction will obtain sampling data (state, action, reward, whether it is over, the logarithm of action probability), which is stored in the data pool. If the expected cumulative reward obtained by the strategy is convergent in the interaction, the training can be stopped and the output is an action network, which can be used directly.
3. After collecting a batch of sequence data in the data pool, it will be input into the trainer. According to the parameters of the current model and the effect of interaction, the trainer calculates the loss of the action network, evaluation network, and output strategy distribution based on the PPO method. Based on the stochastic gradient descent, the neural network parameters are updated, which makes the strategy more likely to receive high rewards.
4. The updated strategy is to be input into the simulator for interaction again. Loop this process until the reward converges or reaches a pre-coded maximum number of interactions, and output the action network.

Based on the PPO algorithm, the decision tasks of a single agent are iterated and optimized in the simulator. When extended to multiple UAVs, corresponding adjustments and changes need to be made. In multi-UAV systems, there are usually multiple homogeneous or heterogeneous UAV units. It would be inappropriate to use one network to output the decisions for all units, which would make the algorithm more difficult to run. The first is that all unit action space combinations will grow exponentially, making it difficult to solve large-scale problems. Second, the decision is a multiple UAVs distributed decision. Communication and information acquisition between UAVs have limitations and costs, and a completely centralized approach does not meet the needs of distributed decisions.

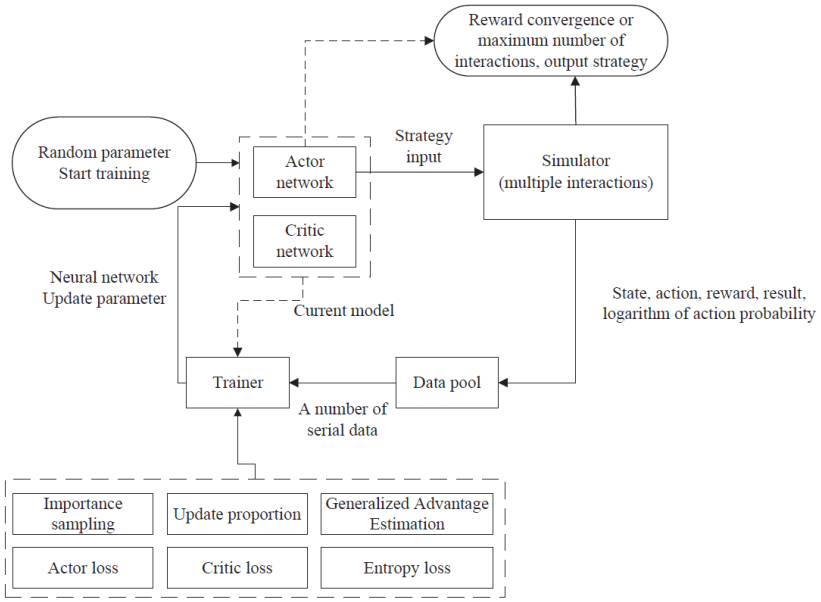

Figure 2. PPO algorithm interaction process.

Therefore, the method used in this paper is to establish a separate PPO algorithm model for each UAV. During the training process, the data of each PPO algorithm model is stored and trained separately. After the data converge, the policy network is deployed to each corresponding UAV unit for decisions. This method is Independent PPO in multiagent reinforcement learning. Figure 3 shows the distributed jamming strategy generation method for multiple UAVs based on the IPPO learning algorithm:

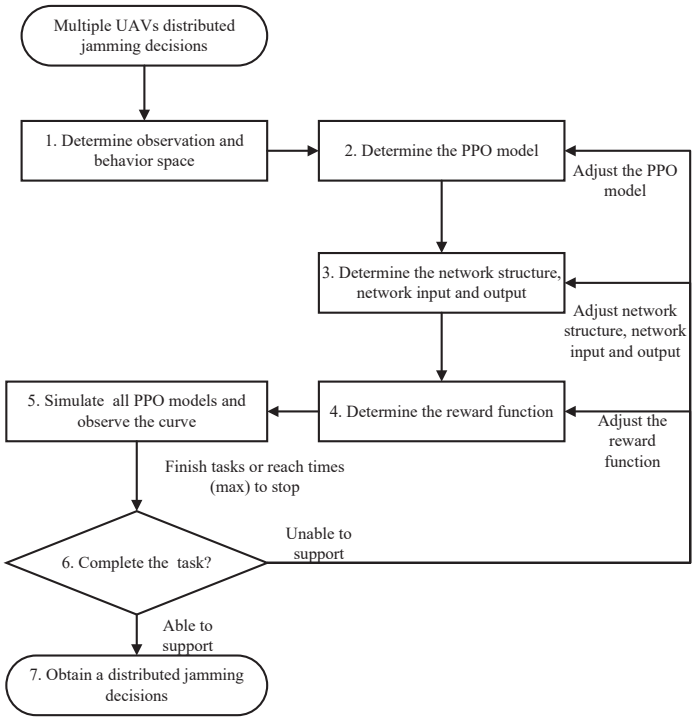

Figure 3. Multiple UAVs distributed jamming decisions based on IPPO.

As shown in the flowchart in Figure 3, a new effective model is obtained based on the IPPO learning algorithm for distributed jamming decisions for multiple UAVs.

1. Determine the observational and behavioral space for each platform.

First, the corresponding observation space and behavior space should be constructed according to the agent model of the UAVs. For the observation space, the feature vectors of the observation need to be constructed to provide sufficient and less redundant observation space for decision-making; the behavior space is to consider the discrete and continuous of the original behavior space.

2. Match the platform with the appropriate PPO model based on its observational and behavioral space.
   After defining the observation and behavioral space of the platform, you can choose whether to use a shared PPO model for units with the same observation and behavioral space. The advantage of using a shared model is that its training data grows multiplicatively, which is conducive to the training and convergence of the model. The disadvantage is that the shared model shows a more consistent behavior, and if you need to show units with different strategies, you need to pair them with a separate PPO model and use your own data for training.
3. Determine the network structure, inputs, and outputs of the network for each PPO model.
   There are two network architectures based on the type of data transmission. The first is a convolutional neural network (CNN network), which is mainly used to transmit some two-digit data. The second is a fully linked network, which is mainly used to transmit one-bit data, such as numbers, vectors, etc. Then the corresponding network structure is determined according to the network input and output of each PPO model.
4. Determine the reward function for each PPO model.
   A corresponding reward function, or target task, needs to be set for each UAV. When the UAV completes the task, it will give positive reward feedback. When the task is not completed, it will give negative reward feedback. If the task is not performed, a zero value is given.
5. Load all PPO models for simulation training and observe the training curves.
   After the reward function is determined, all PPO models are loaded for simulation training. The simulation training is stopped when multiple UAVs complete the coordinated task or when the maximum number of training fields is reached. Based on the effect of simulation training, the training curve of the learning algorithm is combined to determine whether the reward function converges or not.
6. Determine whether the current strategy can support the completion of the UAV coordination mission.
   The reward function converges, then a valid multiple UAV distributed jamming decision model is obtained. If the reward function does not converge, the model does not support the completion of the task. It is necessary to readjust a certain stage of the training process. The adjustment order is suggested to be from adjusting the reward function, adjusting the network structure and the input/output of the network, and adjusting the PPO model match. In the situation where the amount of data is reached, the training does not reach the expected expectations, usually because the task is too difficult to explore. At this point, the reward function can be adjusted to add some guide rewards to make the model more likely to converge. It is also considered whether the current feature vectors are complete and redundant, whether the action space is reasonable, and whether the network structure is too big or too small, and then the network model is adjusted. Finally, if there is an issue of homogeneity in the strategy, is it caused by model sharing and does it need to be reduced or increased in the configuration of model sharing. The operation is repeated until a valid decision model can be obtained.
7. Obtain a coordinated decision-making model for formations.
   If it was tested well in the previous step, the coordinated decision-making model can be adopted as the output of the multi-agent training module.

4. Simulation Electromagnetic Scenario Modeling

In order to prove the IPPO learning algorithm, multiple UAVs are trained in the electromagnetic countermeasures scenario. Since training in a real environment may cause unnecessary losses, it is necessary to build a simulator to prove the IPPO learning algorithm.

In the simulated domain scenario, both sides of the electromagnetic countermeasure scenario are modeled. Assuming that the multiple UAVs are Red, they mainly attack the hostile radar station (training side); the hostile radar station is Blue, which acts as the defensive side (accompanying side) in the simulation scenario. As shown in Figure 4, Blue deployed a total of 4 radar stations, and each radar station is equipped with an air defense position, which can defend a circular area with a radius of 350 km. The four radar stations are distributed around the command and provide dead-end detection of the surrounding area. Red has a total of 4 UAVs in stock, starting the incursion from 500 km from the hostile radar. Each UAV is equipped with passive detection equipment and radar jamming equipment, but the UAV can only activate one function at the same moment due to its real physical properties. That is, the radar position detection and radar signal jamming possessed by the UAV are mutually exclusive behaviors.

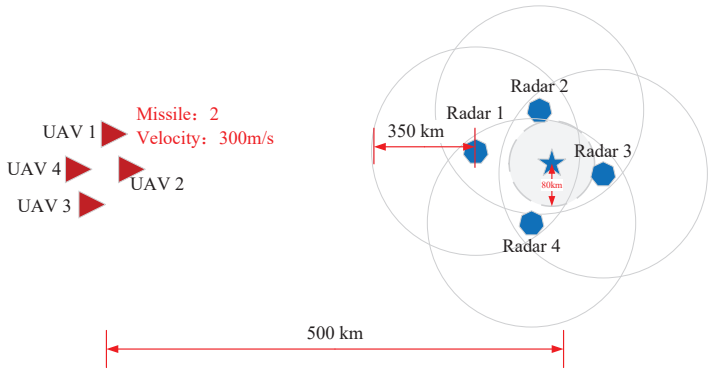

Figure 4. The electromagnetic countermeasure scenario.

As shown in Figure 5, when the UAV is in the perceptual (detection) state, it can obtain the coordinate information of the Blue ground radar through the passive detection equipment it carries. However, due to the influence of the working model of passive detection equipment, UAVs can only detect the radar information that is currently in the boot state. The unpowered radar does not have a radiation source, so it cannot be detected by the UAV.

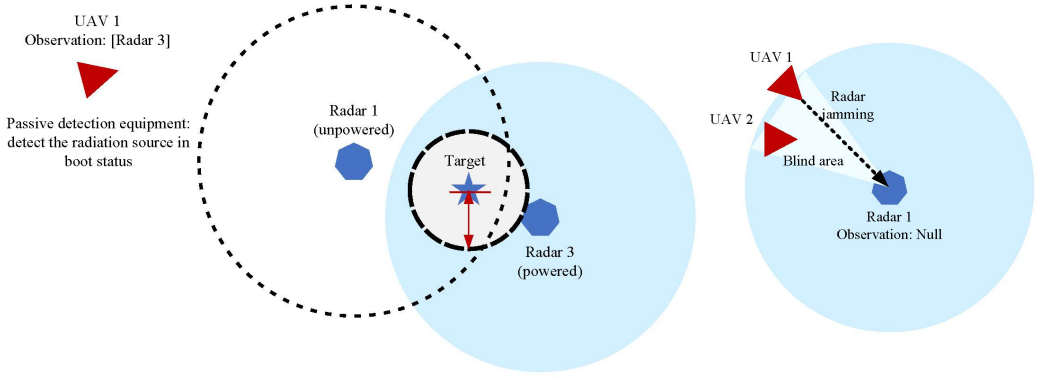

Figure 5. Passive detection scenario and radar jamming scenario.

When the UAV is in a radar jamming state, it exerts an oppressive jamming effect on the detection signal of the hostile radar through the jamming equipment it carries. In the circular detection area of hostile radar, a sector detection blind area γ is created. In this sector blind area, Red will not be detected by radar. Due to the directionality of the

electromagnetic wave emitted by radar jamming, the UAV can only play a role in jamming hostile radar within a certain range.

In the real battleground, the attacker will formulate the corresponding combat strategy after observing the defense strategy of the defender. Therefore, in the simulated electromagnetic countermeasure scenario, when Blue detects the intrusion target, it will quickly notify the equipped air defense position to defend. After observing the defense strategy, the Red UAV looks for enemy tactical vulnerabilities and develops a combat strategy. As in Figure 6, based on multiagent reinforcement learning, multiple UAVs coordinate through a distributed jamming strategy. Some UAVs perform radar jamming to protect teammates from hostile radar in order to destroy hostile targets.

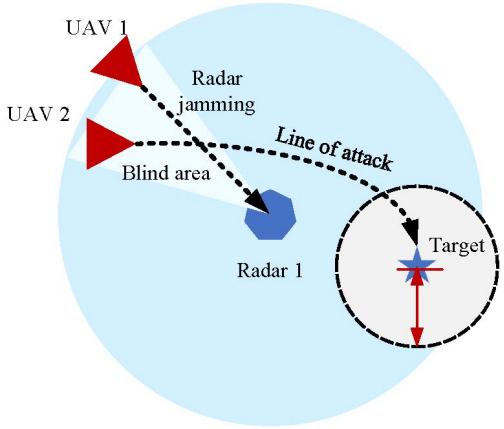

Figure 6. The coordinated penetration scenario.

In the countermeasure scenario, the deductive termination condition exists in two situations: Red wins or Blue wins. The rule of termination conditions are shown in Table 1. If Red UAVs destroy the Blue ground headquarters without being completely destroyed, Red wins. If all Red UAVs are destroyed, or the ammunition of all UAVs is zero and the Blue headquarters is not destroyed, Blue wins.

Table 1. Rule of Termination Conditions.

Winning Side	Termination Conditions
Red	The UAVs destroy the target
Blue	All UAVs are destroyed or the ammunition of all UAVs is zero

5. Experiments and Results

Based on the multiple UAVs countermeasure scenario designed in Section 4, a server with i9-9900KF CPU (3.60 GHz) and GTX2080Ti GPU memory 64 G is used as the training equipment to build a simulator to generate training data.

After 18,000 epochs of continuous training under the simulator, the algorithm returns converged and the training is stopped. The total time is 16 h, 42 min, and 59 s. In the training process, the changes in Loss and Reward are shown in Figures 7 and 8.

At the beginning of training (2 epochs), the UAVs in the exploration phase will be destroyed because they are in the Blue radar detection range. As shown in the left of Figure 9, the UAVs have been in the Blue radar information field for a long time, and the Blue line points to the UAV that has been locked by the Blue radar. In the right of Figure 9, it can be seen that the UAV with id 1 has been destroyed.

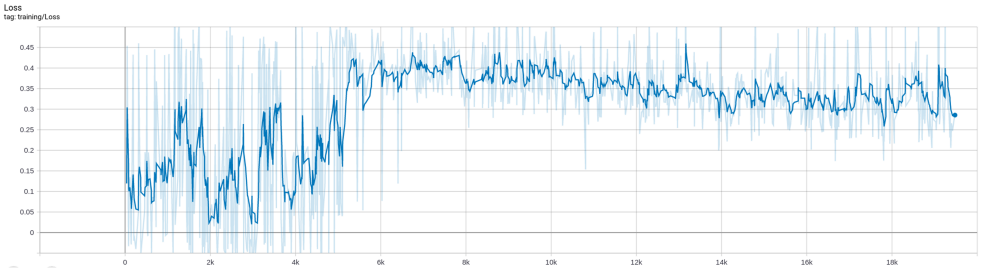

Figure 7. Change of Loss during training.

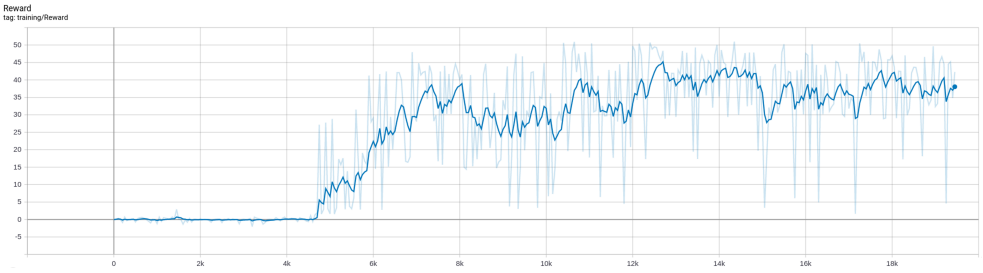

Figure 8. Change of Reward during training.

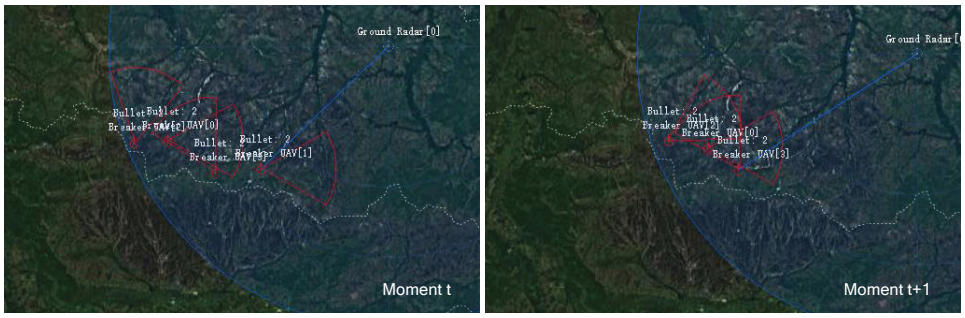

Figure 9. The UAV trained for 2 epochs is locked and destroyed by the Blue radar.

By the 5000th epoch of training, the UAVs have learned to avoid being locked by radar and destroyed by turning on their jamming radar to cause a Blue radar information gap. As shown in Figure 10, the UAV is locked by the radar at moment t. At moment $t+1$, the UAV can unlock itself from the Blue radar by turning on its own jamming radar to cause a gap in the Blue radar scan. By adjusting the operating mode of their radar to avoid being destroyed, they can obtain a negative reward value. As can be seen from Figure 8, the reward function score increases at the 5000th epoch.

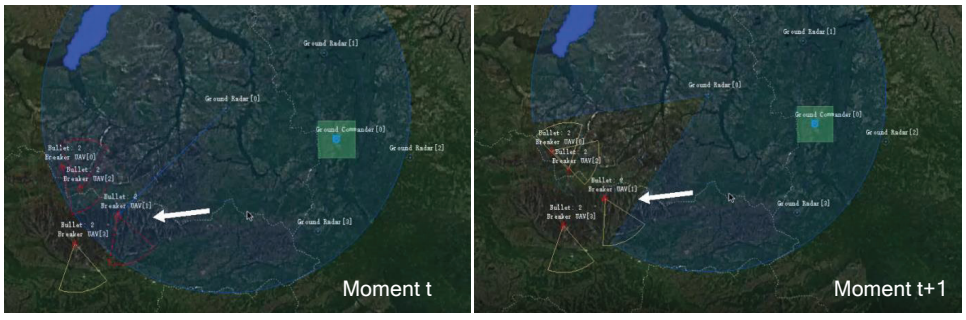

Figure 10. The UAV unlocks by jamming the enemy radar at the 5000th epoch of training.

By the 6000th epoch of training, the UAVs have learned coordinated operations to cover themselves and their teammates in penetrating the radar range. As shown in Figure 11,

two UAVs choose to turn on the radar jamming mode, and the other two turn on the radar perception mode. The jamming UAVs in the rear create an interference deduction environment for the perceptual UAVs, so as to ensure that the perceptual UAVs are not locked by the enemy radar and destroyed.

By the 11,000th epoch of training, the UAVs have learned to destroy enemy ground targets by adjusting their radar modes. As shown in Figure 12, the UAV turns on its jamming radar (the yellow sector in the figure) when it is not within the distance of attacking the enemy ground headquarters, to sneak into the enemy radar detection range without being detected. When the enemy radar enters within attack range, the UAV quickly switches to the radar guidance mode, to complete the missile launch and destroy the enemy ground headquarters. At this time, the model reaches the highest score value. As can be seen from Figure 8, when the model is trained to the 11,000th epoch, the score has converged to the highest score of 50.

Figure 11. The UAVs execute penetration task by cooperating at the 6000th epoch of training.

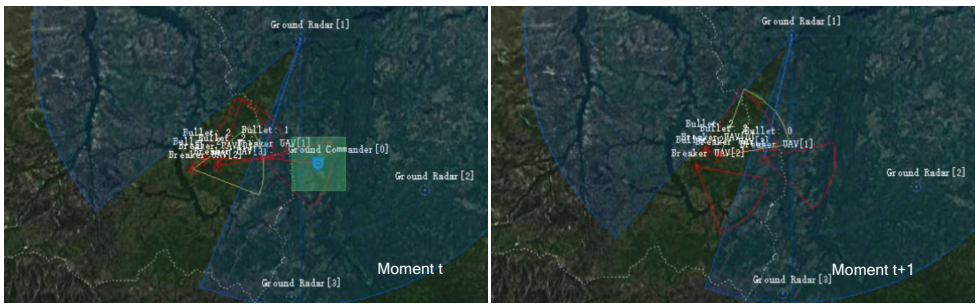

Figure 12. The UAV successfully destroys the enemy headquarters at the 11,000th epoch of training.

In summary, the model score is rapidly improved at the 5000th epoch of training, By the 12,000th epoch, the highest model score converges to about 50 and reaches the highest reward score under the scenario. Thus, the result proves the correctness and effectiveness of the simulation data in the training process.

6. Conclusions

In this paper, we study the application of PPO, a reinforcement learning method proven effective for game strategy generation, in the multiagent distributed airborne electromagnetic countermeasures domain. To easily collect valuable training data, we design a typical electromagnetic countermeasures scenario with multiple UAVs and build a simulator to model all the decision factors and situational deductions needed for multiple UAV coordination. In the simulator, we solve the coordinated strategies of multiple UAVs in an electromagnetic countermeasures environment using IPPO. Experiments and results show our design succeeds in coordinating a group of UAVs equipped with lightweight jamming devices. The coordinated strategies not only handle given jamming tasks against dynamical hostile radar stations but also exhibit heuristic search helping the UAV group find tactical vulnerabilities and improve performance.

Overall, given a sufficiently detailed simulation environment, multiagent reinforcement learning can solve distributed decision problems like airborne electromagnetic countermeasures. The effectiveness of the final strategy depends on whether the simulation granularity represents the real environment's complexities. This allows initial simulation-tested strategies to be further validated in live environments. This simulator-based deep reinforcement learning approach provides a reference under varying strategies, supporting analysis of possible tactical vulnerabilities and advantages.

However, we only explored groups of four UAVs here. Future work could expand the number of UAVs and radar stations, with imbalanced sides and jamming failures or faulty UAVs. Further experiments should evaluate hyperparameter impacts and compare with other coordination approaches to demonstrate performance. This paper focused on UAV radar jamming; the scenario could be extended to more complex and dynamic Red team vs. Blue team settings.

Author Contributions: Methodology, software, writing—original draft preparation, W.R.; software, validation, formal analysis, R.L. (Rong Luo); validation, writing—review and editing, F.Z.; investigation, data curation, writing-review and editing, R.L. (Renwei Luo); conceptualization, supervision and funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zong, Q.; Wang, D.; Shao, S.; Zhang, B.; Han, Y. Research status and development of multi UAV coordinated formation flight control. *J. Harbin Inst. Technol.* **2017**, *49*, 1–14.
2. Xu, Y.; Ruiy, W.; Zhang, T. Review of unmanned aerial vehicle swarm path planning based on intelligent optimization. *Control. Theory Appl.* **2020**, *37*, 2291–2302.
3. Wu, Q.; Wang, H.; Li, X.; Zhang, B.; Peng, J. Reinforcement learning-based anti-jamming in networked UAV radar systems. *Appl. Sci.* **2019**, *9*, 5173. [CrossRef]
4. Li, S.; Jia, Y.; Yang, F.; Qin, Q.; Gao, H.; Zhou, Y. Collaborative Decision-Making Method for Multi-UAV Based on Multiagent Reinforcement Learning. *IEEE Access* **2022**, *10*, 91385–91396. [CrossRef]
5. Ma, J.; Zhou, C.; Zhang, S.; Dong, W.; Zhang, C.; Lin, J. The Design of Simulation System for Multi-UAV Cooperative Guidance. In Proceedings of the IEEE Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control, Kolkata, India, 8–10 January 2016; pp. 1250–1254.
6. Zhu, B.; Feng, H. Building structure simulation system based on BIM and computer model. *J. Sens.* **2021**, *2021*, 8244582. [CrossRef]
7. Kaidi, W.; Chuntao, L.; Peng, C.; Ying, F. Design of real-time and multi-task UAV simulation system based on rapid prototyping. In Proceedings of the 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), Nanjing, China, 12–14 August 2016; pp. 930–936.
8. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
9. 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. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
10. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
11. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354. [CrossRef] [PubMed]
12. Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Dębiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. Dota 2 with large scale deep reinforcement learning. *arXiv* **2019**, arXiv:1912.06680.
13. Källström, J.; Heintz, F. Agent coordination in air combat simulation using multi-agent deep reinforcement learning. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 2157–2164.
14. Soleyman, S.; Khosla, D. Multi-agent mission planning with reinforcement learning. In Proceedings of the AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks, Virtual, 11–12 November 2020.
15. Orr, J.; Dutta, A. Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. *Sensors* **2023**, *23*, 3625. [CrossRef] [PubMed]

16. Yao, W.; de Marina, H.G.; Sun, Z.; Cao, M. Distributed coordinated path following using guiding vector fields. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xian, China, 30 May–5 June 2021.
17. Wang, X.; Baldi, S.; Feng, X.; Wu, C.; Xie, H.; De Schutter, B. A Fixed-Wing UAV Formation Algorithm Based on Vector Field Guidance. In Proceedings of the 2023 IEEE Transactions on Automation Science and Engineering, Auckland, New Zealand, 26–30 August 2023.
18. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
19. Buşoniu, L.; Babuška, R.; Schutter, B.D. Multi-agent reinforcement learning: An overview. *In Innovations in Multi-Agent Systems and Applications-1*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 183–221.
20. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *In Handbook of Reinforcement Learning and Control*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 321–384.
21. 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. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
22. Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LO, USA, 2–7 February 2018.
23. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning, PMLR, Beijing, China, 21–26 January June 2014; Volume 32, pp. 387–395.
24. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
25. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1928–1937.
26. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.