

Article

Not peer-reviewed version

Multi-Radar Trajectory Planning Method Based on Imitation Learning

[Xuchao Gao](#), [Mingqiang Li](#)^{*}, [Kai Guan](#), [Jianjun Ge](#)

Posted Date: 10 March 2026

doi: 10.20944/preprints202603.0704.v1

Keywords: multi-radar; trajectory planning; imitation learning; coverage modeling; electronic countermeasures



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Multi-Radar Trajectory Planning Method Based on Imitation Learning

Xuchao Gao, Mingqiang Li *, Kai Guan and Jianjun Ge

Information Science Academy, China Electronics Technology Group Corporation

* Correspondence: limingqiang14@mails.ucas.ac.cn

Abstract

To address the high computational complexity and insufficient real-time performance of traditional multi-radar trajectory planning methods in complex electromagnetic interference environments, this paper proposes an imitation learning-based trajectory planning method for multi-radar systems. This method designs a trajectory policy neural network architecture based on multiple semantic information. It proposes a training data construction method with coverage rate as the optimization objective. Then the trajectory policy neural network is trained by using an imitation learning algorithm with an auxiliary target. Simulation results show that the proposed method achieves an average coverage rate of 93.95%, and improves the single-step decision efficiency by a factor of 6.7 compared with heuristic-based trajectory optimization methods.

Keywords: multi-radar; trajectory planning; imitation learning; coverage modeling; electronic countermeasures

1. Introduction

With the continuous evolution of informatized warfare and intelligent combat forms, electromagnetic space has become another key operational domain in modern warfare following land, sea, air, and space [1]. Radar has been playing a more and more important role in the target tracking field[2]. As important sensors for acquiring battlefield situational information, radar systems face severe challenges to their survivability and mission effectiveness in complex and highly adversarial electronic warfare environments. The traditional operational mode relies on a single large early warning aircraft or centralized radar platform. Due to its prominent target characteristics and concentrated system structure, traditional mode is highly vulnerable to system failure or even complete paralysis once subjected to electronic jamming, anti-radiation strikes, or saturation attacks. These countermeasures make it difficult to meet the operational requirements for sustained and stable perception in high-threat environments.

In recent years, with the rapid development of unmanned aerial vehicle (UAV) technology, communication technology, and distributed cooperative control theory, distributed multi-radar systems have gradually become a research hotspot. Recent information battlefield scenarios urge modern radars function as a radar network to overcome the information uncertainties and limitations[3]. By distributing sensing, processing, and decision-making capabilities across multiple low-cost, cooperative unmanned platforms, multi-radar systems offer significant advantages such as target dispersion, high redundancy, and strong system survivability[4]. They show greater survivability, flexibility, and environmental adaptability in complex electromagnetic adversarial environments. However, the operational effectiveness of multi-radar systems depends on a large extent on the trajectory planning and cooperative maneuvering capabilities of multi-radar members in dynamic threat environments. One of the important challenges in this field is how to achieve effective evasion of enemy threats while satisfying radar detection performance constraints.

To address the problem of efficient online trajectory planning for multi-radar systems in jamming environments, this paper proposes an imitation learning-based trajectory planning method

for multi-radar systems. First, a trajectory policy representation neural network architecture based on multi-semantic information fusion is designed. This architecture unifies the encoding of regional coverage status, target motion information, and radar own maneuvering state. Second, a training data construction method with coverage rate as the optimization objective is proposed. By establishing a mathematical optimization model for trajectory planning and solving it offline, an expert demonstration dataset in the form of state-action pairs is constructed. Finally, an imitation learning algorithm oriented toward coverage tasks is proposed to train the trajectory policy neural network. The network enables the policy model to rapidly infer and output heading control commands during the online phase, and significantly improves the real-time performance of trajectory planning while maintaining coverage performance. Simulation results demonstrate that the proposed method achieves an average coverage rate of 93.95%, and archives a 6.7-fold improvement in single-step decision efficiency compared with heuristic-based trajectory optimization methods.

2. Current Research Status

Regarding the trajectory planning problem of multi-radar systems, many scholars have conducted extensive research. In terms of mathematical model construction, most existing studies on multi-radar system trajectory planning focus on mathematical modeling for optimal detection capability in non-jamming environments. However, research addressing toward electronic warfare, particularly active jamming environments, remains relatively limited. For example, Yan et al. designed a UAV path planning approach considering attenuated regional coverage[5]; Lu et al. improved the multi-target tracking capability of airborne radar systems through path planning [6]; Duan et al. solved the path planning problem for multiple reconnaissance-strike integrated UAVs [7]; Zhang et al. investigated path planning for UAV area coverage reconnaissance missions in interference-free conditions [8].

In terms of trajectory planning model solving algorithms, existing research mostly adopt heuristic algorithms [9] or deep reinforcement learning algorithms. Heuristic methods have the advantages of intuitive modeling and ease of implementation, enabling the acquisition of feasible solutions or even near-optimal solutions to a certain extent. However, these algorithms generally suffer from high computational complexity, insufficient real-time performance, and limited adaptability to environmental changes. Especially in high-dimensional, highly adversarial, and dynamically evolving electronic warfare scenarios, they struggle to meet the demands for rapid online decision-making. Among these, Zhang et al. proposed the NSBOA algorithm to optimize UAV paths [10]; Jiang et al. adopted the kite algorithm to design UAV obstacle avoidance path strategies in 3D space [11]; Wei et al. used the sparrow search algorithm to study trajectory planning in 3D space [12]; Liu et al. adopted the particle swarm optimization algorithm to plan routes for maneuvering reconnaissance platforms [13]; Besada-Portas et al. applied genetic algorithms to multi-UAV path planning [14]; Shi et al. used particle swarm optimization and cyclic minimization methods to optimize routes under multi-target tracking [15]; Zhang et al. used a hybrid stochastic simulation and genetic algorithm to optimize the airborne radar network problem [16]; Yuan et al. used diffusion model to solve radar selection problem[17].

In recent years, reinforcement learning is able to learn optimal strategies through interaction with the environment without requiring precise modeling. It has become an important method for multi-UAV trajectory planning [18]. For instance, Tang et al. did a research on UAV autonomous trajectory planning based on prediction information in crowded unknown dynamic environments[19]; Li et al. did a research on trajectory planning of autonomous vehicles in constrained spaces[20]; Xu et al. used the DQN method to study trajectory planning in a dynamic environment with complex 3D low-altitude terrain [21]; Gao et al. used the DDPG method to provide a trajectory planning method that avoids actual terrain [22]; Xu et al. used PPO to solve the path planning problem for UAV swarms in GPS and communication denial environments [23]; Wang et al. used MADDPG to study the UAV swarm pursuit decision-making problem in air combat

environments [24]. Although reinforcement learning methods have strong potential in complex environmental perception and autonomous strategy learning, their training process typically relies on extensive online interaction. This leads to problems such as high sample acquisition costs, long training cycles, and insufficient stability. Consequently, it limits their engineering application in multi-radar trajectory planning tasks with high adversarial and strong real-time requirements. Therefore, how to improve the training efficiency of trajectory policy networks has become an urgent problem to be solved.

Based on the current research, this paper's key innovations include: 1) the influence of active electronic jamming on the detection capability of multi-radar systems is taken into consideration. This makes the proposed trajectory planning method more suitable for complex confrontation environments; 2) this paper designs a trajectory policy representation neural network architecture based on multi-semantic information fusion. The architecture uses the powerful nonlinear mapping capability and fast online inference characteristics of deep neural networks to solve the problem of insufficient real-time solving efficiency of traditional heuristic algorithms; 3) this paper proposes a supervised training approach that uses the trajectory planning method of traditional heuristic algorithms as the imitation target to train the trajectory policy neural network and enhance its fast convergence capability.

3. System Modeling and Problem Formulation

Consider a cooperative detection scenario of a multi-radar system in a two-dimensional plane. The multi-radar system consists of N UAV radars that are spatially distributed over a wide area and capable of free maneuvering. The scenario also includes M target nodes and J fixed-interval support jammers with noise suppression jamming. The key coverage area is a rectangle with side lengths of $R_x \times R_y$, where R_x and R_y are the region side length along the x -axis and y -axis directions, respectively. The key coverage area is discretized into $L_1 \times L_2$ sampling points in the two-dimensional e , where L_1 and L_2 are the numbers of discrete sampling points along the x -axis and y -axis directions of the two-dimensional region, respectively. This point set Ω is denoted as

$$\Omega = \{r_1, r_2, \dots, r_{L_1 \times L_2}\}, r_l \in R^2, l = 1, \dots, L_1 \times L_2, \quad (1)$$

Each UAV radar maneuvers within the restricted safe airspace A_r . The radars adjust its trajectory to achieve continuous coverage of the key area and effective detection of targets. The multi-radar system starts from time 0 and makes heading offset angle decisions at intervals of Δt to accomplish the tasks of key area coverage and target detection. For convenience, any discrete time instant $t \in \Delta t (t = 0, 1, \dots, T)$ is referred to simply as time t .

Let $P^t = (P_1^t, \dots, P_N^t) \in R^{2 \times N}$ denote the position matrix of N UAV radars in the XOY horizontal plane at time t , where $P_n^t \in R^{2 \times 1}$ denotes the position of the n -th ($n = 1, \dots, N$) UAV radar; $\theta^t = (\theta_1^t, \dots, \theta_N^t) \in R^{1 \times N}$ denotes the heading vector of N UAV radars at time t , where $\theta_n^t \in R$ denotes the heading of the n -th ($n = 1, \dots, N$) UAV radar; $Q^t = (Q_1^t, \dots, Q_M^t) \in R^{2 \times M}$ denotes the position matrix of M targets in the XOY horizontal plane at time t , where $Q_m^t \in R^{2 \times 1}$ denotes the position of the m -th ($m = 1, \dots, M$) target; $Z = (Z_1, \dots, Z_J) \in R^{2 \times J}$ denotes the position matrix of J stand-off jammers in the XOY horizontal plane, and Z_j denotes the position of the j -th ($j = 1, \dots, J$) jammer.

The trajectory decision problem of the multi-radar system can be described as computing the radar heading offset angle $\Delta\theta^t = (\Delta\theta_1^t, \dots, \Delta\theta_N^t) \in R^{1 \times N}$. This problem is based on the positions of UAV radars, targets, and jammers at time t , so as to compute the position of the UAV radar at time $t + 1$, i.e.,

$$P_n^{t+1} = P_n^t + v\Delta t * \begin{bmatrix} \cos(\theta_n^{t+1}) \\ \sin(\theta_n^{t+1}) \end{bmatrix}, n=1, \dots, N, \quad (2)$$

$$\theta_n^{t+1} = \theta_n^t + \Delta\theta_n^t, n=1, \dots, N, \quad (3)$$

where v is the speed of the UAV, Δt is the time interval from time t to $t+1$, $\Delta\theta_n^t \in [-\Delta\theta_{max}, \Delta\theta_{max}]$ denotes the heading offset angle of the n -th ($n = 1, \dots, N$) UAV radar at time t , and $\Delta\theta_{max}$ denotes the maximum offset angle within the time interval Δt .

4. Trajectory Planning Method Based on Imitation Learning

4.1. Trajectory Policy Representation Neural Network Architecture Based on Multi-Semantic Information Fusion

In the multi-radar trajectory planning task under jamming environments, the multi-radar system needs to rapidly output multi-radar heading control commands based on the system state at time t . This state not only contains the maneuvering information of the radar itself, but is also constrained by multiple factors such as regional coverage distribution, target motion trends, and jamming effects at the same time. It shows significant polysemy and heterogeneity. There are significant differences among different types of information in terms of spatial structure, temporal correlation, and physical meaning. If taking a unified encoding approach, it will be difficult to fully extract the key features highly relevant to trajectory decision-making. To this end, this paper designs a trajectory policy neural network architecture based on multi-semantic information fusion. The mathematical model of the trajectory policy network is expressed as

$$\widehat{\Delta\theta}^t = f(s^t; w), \quad (4)$$

where w is the neural network parameter, $\widehat{\Delta\theta}^t \in R^N$ is the heading offset angle output by the neural network, and $s^t = \{s_{all_cov}^t, s_{single_cov}^t, s_{uav}^t, s_{env}^t\}$ is the multi-semantic information at time t , where $s_{all_cov}^t \in \{0,1\}^{L_1 \times L_2}$ denotes the multi-radar coverage information, $s_{single_cov}^t \in \{0,1\}^{N \times L_1 \times L_2}$ denotes the single-radar coverage information, $s_{uav}^t = \begin{pmatrix} P^t \\ \theta^t \end{pmatrix} \in R^{3 \times N}$ denotes the instantaneous state information of the multi-radar system, and $s_{env}^t \in R^{K \times 2 \times M}$ denotes the target perception information of the multi-radar system over the preceding K time instants, i.e.,

$$s_{env}^t = \begin{pmatrix} Q^t \\ Q^{t-1} \\ \vdots \\ Q^{t-K+1} \end{pmatrix}. \quad (5)$$

4.1.1. Computation of Multi-Semantic Information

At each time instant t , the multi-semantic information s^t serving as input to the neural network needs to be dynamically computed. Among these, s_{uav}^t and s_{env}^t are related to the radar's own motion state information and target detection results. They are relatively straightforward to obtain. The following focuses on the construction methods for $s_{single_cov}^t$ and $s_{all_cov}^t$.

Let $P_{d,n}^t(r_l)$ denote the detection probability of the n -th ($n = 1, \dots, N$) UAV radar for a target at spatial position r_l ($l = 1, \dots, L_1 \times L_2$). Computed according to CFAR detection theory, $P_{d,n}^t(r_l)$ can be expressed as

$$P_{d,n}^t(r_l) \approx \frac{1}{2} \operatorname{erfc} \left(\sqrt{-\ln P_{fa}} - \sqrt{\operatorname{SNR}_{n,l}^t} \right), \quad (6)$$

where P_{fa} is the false alarm rate, and erfc is the complementary error function, i.e.,

$$\operatorname{erfc}(u) = 1 - \frac{2}{\sqrt{\pi}} \int_0^u e^{-v^2} dv, \quad (7)$$

and $\operatorname{SNR}_{n,l}^t$ denotes the signal-to-interference-plus-noise ratio of the target echo at position r_l received by radar n , i.e.,

$$\operatorname{SNR}_{n,l}^t = 10 \log \frac{P_{n,l}}{\sigma_w^2 + \sum_{j=1}^J P_{n,j}}, \quad (8)$$

where σ_w^2 is the noise power, $P_{n,l}$ is the target echo power at position r_l received by radar n , and $P_{n,j}$ is the jamming power from the j -th stand-off support jammer received by radar n .

The coverage $\mathbf{s}_{single_cov}^t[n, l_1, l_2]$ of the n -th ($n = 1, \dots, N$) UAV radar at spatial position r_l is expressed as

$$\mathbf{s}_{single_cov}^t[n, l_1, l_2] = \begin{cases} 1, & \text{if } P_{d,n}^t(r_l) \geq 0.5 \\ 0, & \text{if } P_{d,n}^t(r_l) < 0.5 \end{cases}, l = l_1 \times l_2. \quad (9)$$

Assuming the multi-radar system adopts a multi-radar cooperative detection mechanism, the joint detection probability for a target at sampling point r_l can be expressed as

$$P_d^t(r_l) = 1 - \prod_{n=1}^N (1 - P_{d,n}^t(r_l)), \quad (10)$$

and the coverage $\mathbf{s}_{all_cov}^t[l_1, l_2]$ of the multi-radar system at spatial position r_l is expressed as

$$\mathbf{s}_{all_cov}^t[l_1, l_2] = \begin{cases} 1, & \text{if } P_d^t(r_l) \geq 0.5 \\ 0, & \text{if } P_d^t(r_l) < 0.5 \end{cases}, l = l_1 \times l_2. \quad (11)$$

4.1.2. Multi-Semantic Information Feature Fusion

The policy network architecture is shown in Figure 1. The trajectory policy neural network consists of a coverage information fusion module, a motion and temporal information fusion module, and a feature fusion module. The coverage information fusion module and the motion and temporal information fusion module perform fusion on different types of state information respectively. After fusion, the results are uniformly input into the feature fusion module, which ultimately outputs the heading offset angle control commands for the multi-radar system.

The roles of the different modules of the policy network are as follows:

1. Coverage information fusion module f_1 : produces output $t_1 \in R^{128}$ by performing convolutional operations on $\mathbf{s}_{all_cov}^t$ and $\mathbf{s}_{single_cov}^t$.

$$t_1 = f_1(\mathbf{s}_{all_cov}^t, \mathbf{s}_{single_cov}^t; w_1), \quad (12)$$

2. Motion and temporal information fusion module f_2 : produces output $t_2 \in R^{128}$ by performing embedding and LSTM operations on \mathbf{s}_{uav}^t and \mathbf{s}_{env}^t .

$$t_2 = f_2(\mathbf{s}_{uav}^t, \mathbf{s}_{env}^t; w_2), \quad (13)$$

3. Feature fusion module f_3 : fuses the outputs t_1 and t_2 of the preceding two modules to obtain the fused feature vector $t_3 \in R^{256}$.

$$t_3 = f_3(t_1, t_2; w_3), \quad (14)$$

4. Action generation module f_4 : subsequently applies fully connected layer computation to t_3 , and uses the tanh activation function to constrain the output values to the interval $[-1, 1]$. Then it is multiplied by the preset maximum offset angle $\Delta\theta_{max}$. So the generated N -dimensional action vector $\widehat{\Delta\theta}^t$ is constrained within the permissible range. Thus, the final heading deflection angle $[\widehat{\Delta\theta}_1^t, \widehat{\Delta\theta}_2^t, \dots, \widehat{\Delta\theta}_N^t]^T$ assigned to each radar is determined.

$$\widehat{\Delta\theta}^t = f_4(t_3; w_4), \quad (15)$$

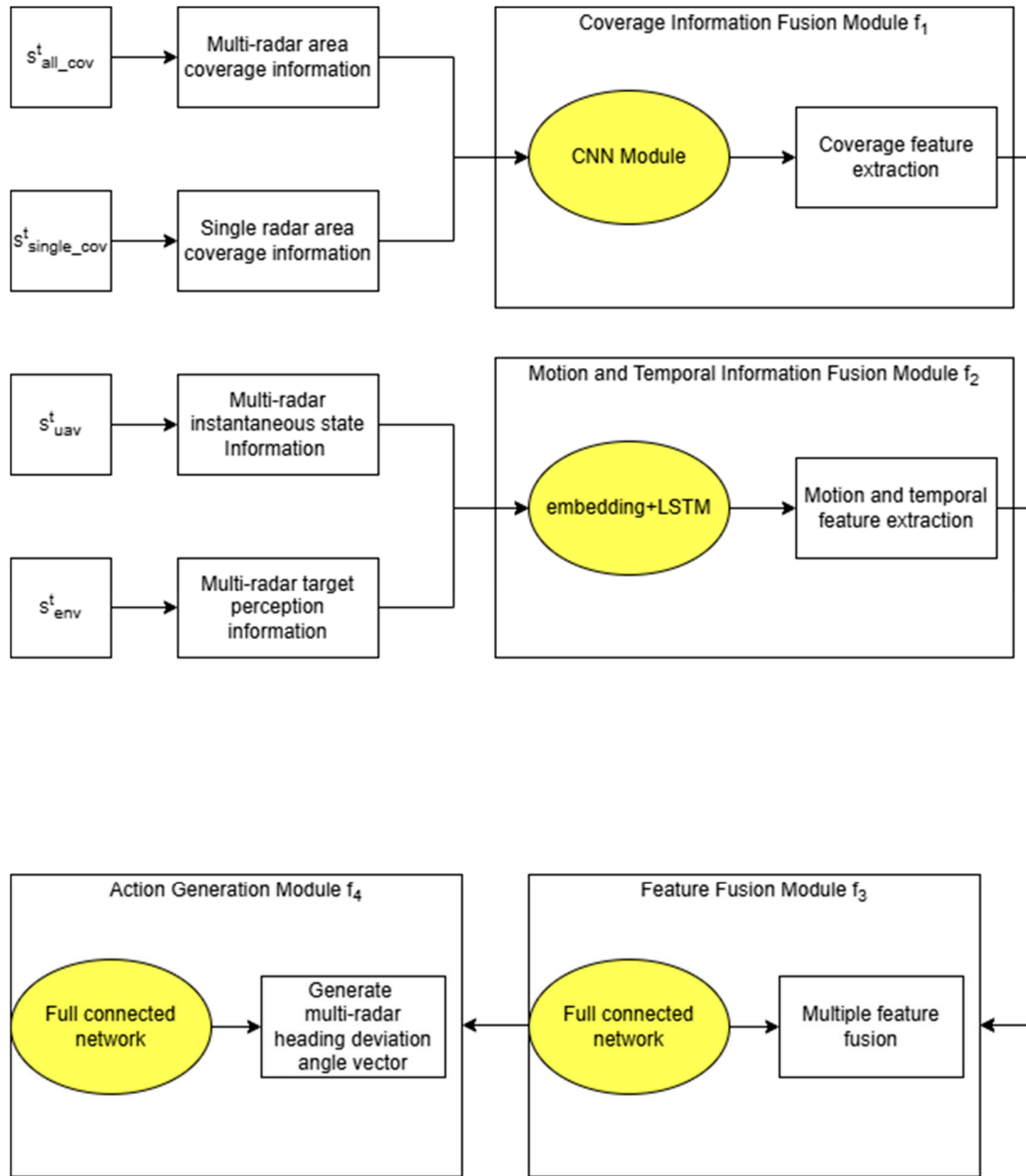


Figure 1. policy network architecture.

4.2. Trajectory Policy Network Training Based on Imitation Learning

To address the problem of fast training of the trajectory policy network, this paper proposes using the trajectory planning method of traditional heuristic algorithms as the imitation target. And it employs a supervised approach to train the trajectory policy neural network.

The trajectory planning method of traditional heuristic algorithms generally solves the following mathematical model, i.e.,

$$\begin{aligned} & \max C^{t+1} \\ \text{s.t.} & \begin{cases} -\Delta\theta_{\max} \leq \Delta\theta_n^t \leq \Delta\theta_{\max}, n=1, \dots, N, t=0, \dots, T-1 \\ P_n^{t+1} \in A_r, n=1, \dots, N, t=0, \dots, T-1 \\ P_n^{t+1} = P_n^t + v\Delta t \begin{bmatrix} \cos(\theta_n^t + \Delta\theta_n^t) \\ \sin(\theta_n^t + \Delta\theta_n^t) \end{bmatrix}, n=1, \dots, N, t=0, \dots, T-1 \end{cases}, \end{aligned} \quad (16)$$

where C^{t+1} denotes the effective coverage rate of the key area at time $t + 1$, i.e.,

$$C^{t+1} = \frac{1}{L_1 \times L_2} \sum_{l=1}^{L_1 \times L_2} I(P_d^{t+1}(r_l) \geq 0.5), \quad (17)$$

where $I(\cdot)$ is the indicator function. Let $\Delta\check{\theta}^t$ be the optimal solution of the above mathematical model. The imitation learning training dataset is then defined as $X = \{(s^t, \Delta\check{\theta}^t)\}$, where t is a discrete sampling time point.

Based on the above dataset, the training of the policy network parameters is achieved by optimizing the following loss function:

$$L_{BC}(w) = \sum_t \|f(s^t; w) - \Delta\check{\theta}^t\|^2, \quad (18)$$

where $f(s^t; w)$ denotes the heading offset angle output by the network, and $\Delta\check{\theta}^t$ is the corresponding optimal heading offset angle obtained from the optimization model. The full primary task training objective, incorporating an L_2 regularization term, is:

$$L(w) = L_{BC}(w) + \lambda \|w\|, \quad (19)$$

where $\lambda > 0$ is a regularization hyperparameter.

To further improve the convergence speed and generalization performance of the trajectory policy network, an auxiliary task is introduced only during the training phase. Specifically, an additional prediction head f_5 is appended after the feature fusion module output t_3 . This module takes t_3 as input and outputs the predicted target positions $\hat{Q}^t = f_5(t_3; w_5) \in \mathbb{R}^{2 \times M}$ at the time t . To decrease the influence of positional scale differences on the loss magnitude, the coordinates of the predicted and ground-truth target positions are normalized by the maximum side length of the key coverage area. The auxiliary task loss is then defined as:

$$L_{aux}(w) = \sum_t \left| \frac{\hat{Q}^t}{\max(R_x, R_y)} - \frac{Q^t}{\max(R_x, R_y)} \right|^2. \quad (20)$$

When the auxiliary task is enabled, the overall training loss becomes:

$$L_{total}(w) = L_{BC}(w) + \alpha L_{aux}(w) + \lambda \|w\|, \quad (21)$$

where $\alpha \in \{0,1\}$ is an auxiliary task flag that enables the auxiliary tasks. It is important that the auxiliary prediction head and its associated loss L_{aux} serve solely as a training-phase regularization to guide feature representation learning. During inference, the auxiliary head is removed, and the network outputs only the heading offset angles. To ensure a fair comparison between the configurations with and without the auxiliary task, the primary task loss $L(w) = L_{BC}(w) + \lambda \|w\|$ defined in Equation (19) is used as the unified evaluation metric in all experiments, regardless of whether the auxiliary task is incorporated during training.

The overall procedure of the imitation training algorithm is presented as follows:

Algorithm 1 Imitation Training Algorithm for Multi-Radar Trajectory Decision-Making Based on Imitation Learning

Input: Number of decision time instants T ; particle swarm optimization solver $solver(\cdot)$; learning rate η ; number of training epochs E ; batch size B ; loss function hyperparameter λ ; auxiliary task flag $\alpha \in \{0,1\}$.

Output: Trajectory policy network parameters w .

Step 1: Training Data Generation

1. Initialize training data $X \leftarrow \emptyset$;
 2. **for** $t = 0$ **to** $T - 1$ **do**
 3. Establish the optimization model shown in (16);
 4. Update $s^t \leftarrow \{s_{all_cov}^t, s_{single_cov}^t, s_{uav}^t, s_{env}^t\}$;
 5. Call the particle swarm optimization solver to obtain $\Delta\check{\theta}^t = (\Delta\check{\theta}_1^t, \dots, \Delta\check{\theta}_N^t) \leftarrow \underset{\Delta\check{\theta}_n^t}{\underset{\Delta\check{\theta}_n^t}{argmax}} \{model(16)\}, n = 1, \dots, N$
 6. Update training dataset $X \leftarrow X \cup \{(s^t, \Delta\check{\theta}^t)\}$;
-

-
7. Execute action $\Delta\check{\theta}^t = (\Delta\check{\theta}_1^t, \dots, \Delta\check{\theta}_N^t)$, update the positions and headings of the multi-radar system according to equations $\theta_n^{t+1} = \theta_n^t + \Delta\check{\theta}_n^t, P_n^{t+1} = P_n^t + v\Delta t * \begin{bmatrix} \cos(\theta_n^{t+1}) \\ \sin(\theta_n^{t+1}) \end{bmatrix}, n = 1, \dots, N;$
 8. $t \leftarrow t + 1;$
 9. **end for**
 10. **return** X

Step 2: Updating Trajectory Policy Network Parameters via Imitation Learning

1. Randomly initialize trajectory policy network parameters w ;
 2. **for** $epoch = 1$ **to** E **do**
 3. Sample a batch from dataset $X: \{(s^i, \Delta\check{\theta}^i)\}_{i=1}^B;$
 4. Compute loss function $L(w) \leftarrow \frac{1}{B} \sum_{i=1}^B \|f(s^i; w) - \Delta\check{\theta}^i\|^2 + \alpha L_{aux}(w) + \lambda \|w\|;$
 5. Backpropagate to compute gradient $g \leftarrow \nabla_w L(w);$
 6. Update parameters $w \leftarrow w - \eta \cdot g;$
 7. **end for**
 8. **return** w
-

5. Simulation Experiments

5.1. Experimental Scenario Description

As shown in Figure 2, yellow represents targets, red represents radars, and purple represents jammers. The targets attempt to penetrate the radar network along the blue arc trajectory. The radar nodes are each equipped with a three-face radar and move within the movable area bounded by the dashed rectangle. They attempt to maximize the coverage of the entire rectangular area to be covered and to track and detect the targets. The experimental hardware configuration consists of an AMD R5-7500F CPU and an Nvidia RTX 3090 GPU, and the experimental environment is Python version 3.12.

In the simulation scenario, the target movement trajectory is arc-shaped, and the variation of target position with time can be expressed by the following equation:

$$\begin{cases} \varphi_m^t = \varphi_m^0 + \Delta t \cdot \phi \\ x_m^t = x_m^0 + r \cdot \cos(\varphi_m^t), \\ y_m^t = y_m^0 + r \cdot \sin(\varphi_m^t) \end{cases} \quad (22)$$

where r and ϕ are the radius and angular velocity of the arc trajectory, taking values of 81.25 km and 4.49°/s, respectively.

The simulation experiment parameters are presented in Table 1. The initial radar information, initial target information, and jammer information in the simulation scenario are provided in Table 2, Table 3, and Table 4, respectively.

Table 1. experiment parameters.

Parameters	Value
Number of radar nodes N	4
Number of target nodes M	4
Number of jammer nodes J	4
Number of decision time instants T	30
Coverage area range	(150km,150km)
Radar antenna gain	30dB
Jammer antenna gain	30dB
Radar flight speed	200m/s
Target radar cross section	16m ²
Movable area range	(30km,30km)

Number of preceding time instants K	5
Radar transmit power	4500W
Stand-off support jammer transmit power	1000W
Maximum radar heading deflection angle	30°

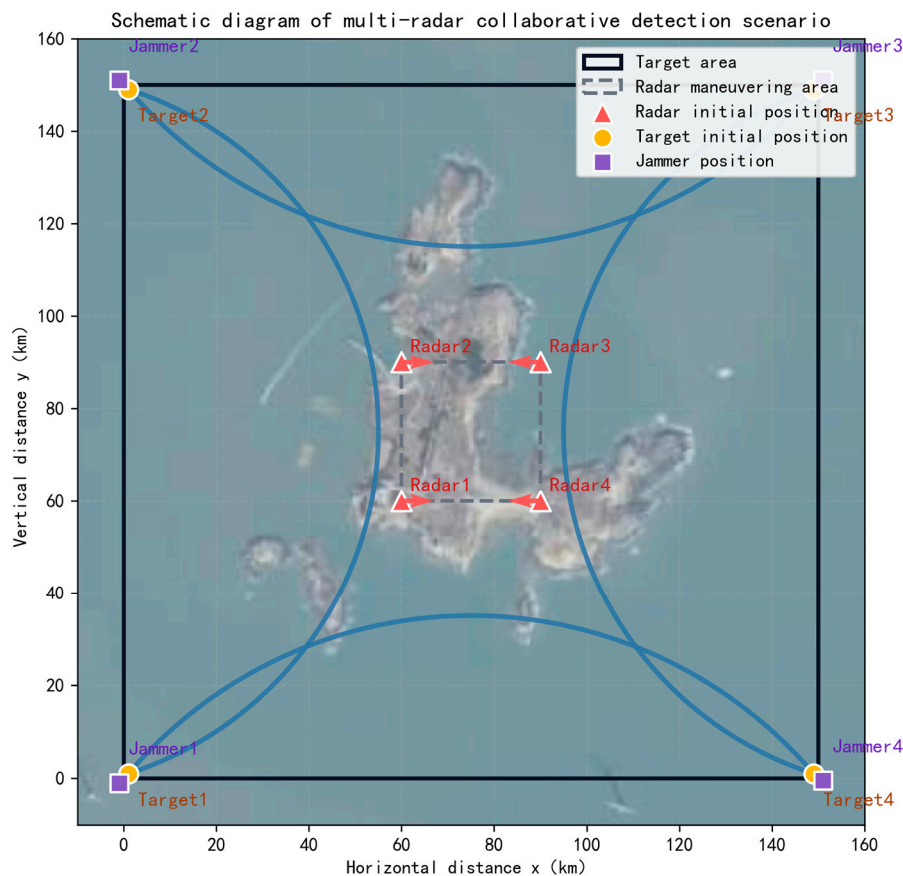


Figure 2. scene diagram.

Table 2. radar initial information.

Radar initial position and heading			
Radar 1	Radar 2	Radar 3	Radar 4
(60km,60km,0°)	(60km,90km,0°)	(90km,90km,180°)	(90km,60km,180°)

Table 3. target initial information.

Target initial position			
Target 1	Target 2	Target 3	Target 4
(0km,0km)	(0km,150km)	(150km,150km)	(150km,0km)

Table 4. jammer information.

Jammer position			
Jammer 1	Jammer 2	Jammer 3	Jammer 4
(0km,0km)	(0km,150km)	(150km,150km)	(150km,0km)

5.2. Training Performance Experiment

In this paper, the particle swarm optimization algorithm is used to solve the single-step trajectory optimization mathematical model. The solutions obtained by this algorithm are used as the

training data for the trajectory policy network. Figure 3 illustrates the variation of the primary task loss function $L(w)$ during the training process under the configurations with and without the auxiliary task. As shown in the figure, both curves exhibit an overall downward trend, decreasing from 0.19 to 0.14. The key difference is that the configuration with the auxiliary task converges more rapidly during the first 400 epochs compared to the configuration without it. The two curves gradually converge after 600 epochs. This shows that incorporating the auxiliary task can accelerate the convergence of the primary task during training.

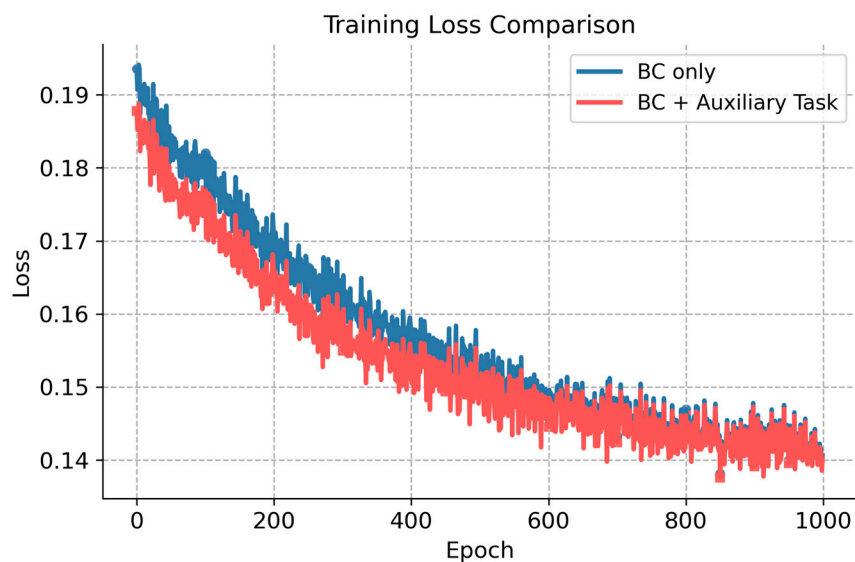


Figure 3. policy network loss function variation chart.

Figure 4 presents the variation of the auxiliary task loss L_{aux} during the training process. The auxiliary task loss exhibits a consistent downward trend, decreasing from 0.35 to 0.13 over 1000 epochs. This indicates that the auxiliary task converges stably and achieves satisfactory performance in target position prediction.

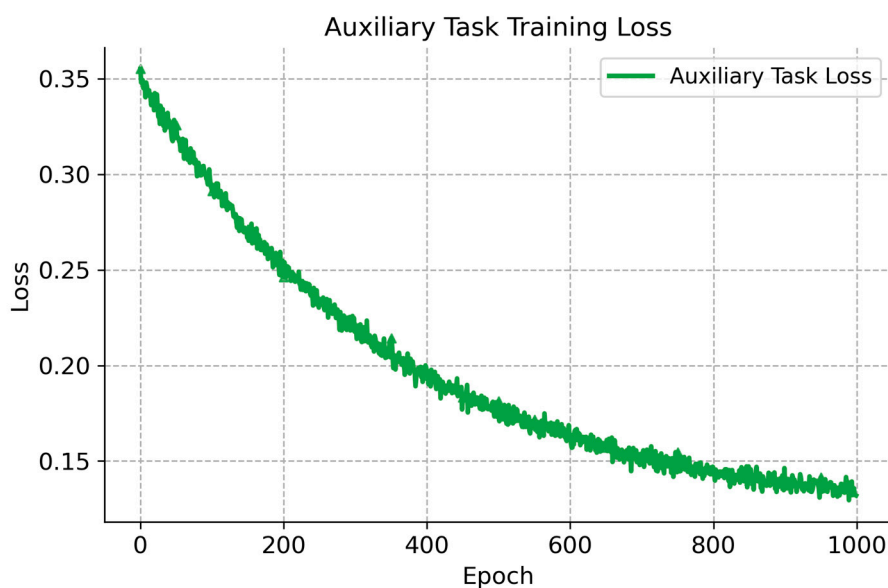


Figure 4. auxiliary task training loss.

Figure 5 shows the similarity between the output of the trajectory policy network and the particle swarm method on random samples. The average similarity across 100 samples can reach 81.1%, as indicated by the gray dashed line in the figure. The lowest similarity of 75.2% appears at the 91st sample, while the highest similarity of 88.0% is achieved at the 53rd sample. It shows good imitation performance of the policy network.

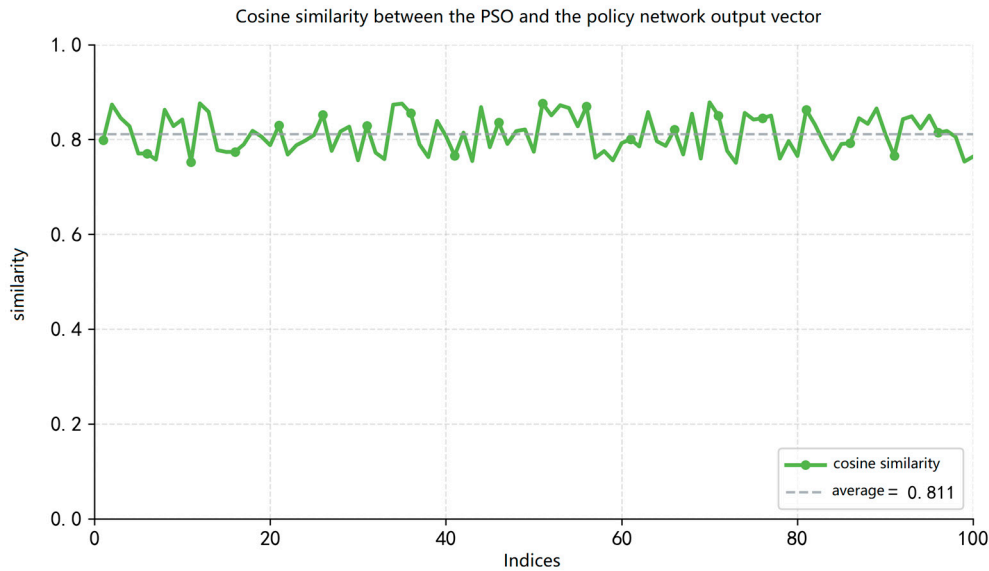


Figure 5. similarity of actions output by trajectory policy network on random samples.

5.3. Inference Capability Experiment

Figure 6 describes the experimental results of the trajectory policy network proposed in this paper and the particle swarm algorithm in terms of coverage capability. In the figure, the gray dashed line represents the variation of the coverage rate of the particle swarm algorithm, and the red solid line represents the variation of the coverage rate of the imitation learning method. During the time period $t = 0 \sim 1$, the coverage rate of both methods decreases from 95.86% to 92.63%. This is because at the initial moment, the radars, targets, and jamming sources are all in a uniformly distributed state. As the radars and targets move forward, their motion is constrained by the radar's own motion constraints and boundary constraints. The overall system coverage rate experiences a phased decline. Subsequently, during the time period $t = 2 \sim 13$, the radars gradually adjust their positions and form an effective deployment. The coverage rate of both methods shows an upward trend, rising from 92.63% to 95.09%. During the time period $t = 14 \sim 18$, the coverage rate stabilizes and fluctuates around 95.00%. After $t = 20$, the coverage rate of both methods oscillates slightly around 93.77%.

From the overall trend, the trajectory policy network is basically consistent with the particle swarm algorithm in terms of coverage rate variation characteristics. The policy network effectively reproduces the decision-making behavior of the particle swarm algorithm. However, in terms of quantitative indicators, the trajectory policy network achieves an average coverage rate of 93.95% across all decision points throughout the entire process. This is lower than the particle swarm algorithm's 94.02%. This difference mainly stems from the inevitable error accumulation effect of imitation learning in sequential decision-making processes.

Figure 7 presents a comparison of the coverage rate between the policy network trained with the auxiliary task and that trained without it. From the overall trend, the policy network trained with the auxiliary task consistently achieves a higher coverage rate than that trained without it throughout the entire decision-making process. Regarding the temporal performance, the coverage rate of the configuration with the auxiliary task is described in Figure 6. The configuration without auxiliary

tasks experiences a sharp drop in coverage from 95.86% to 91.03% at $t = 2$, followed by a steady increase reaching 94.42% at $t = 12$. From $t = 12$ to $t = 19$, the coverage rate fluctuates around 94%. It then begins to decline, reaching its lowest point of 89.27% at $t = 24$. Coverage subsequently recovers to 92.31% in the subsequent steps. The policy network trained with the auxiliary task demonstrates superior coverage performance compared to that trained without it. This indicates that the auxiliary task yields a significant beneficial effect during the training of the trajectory policy network.

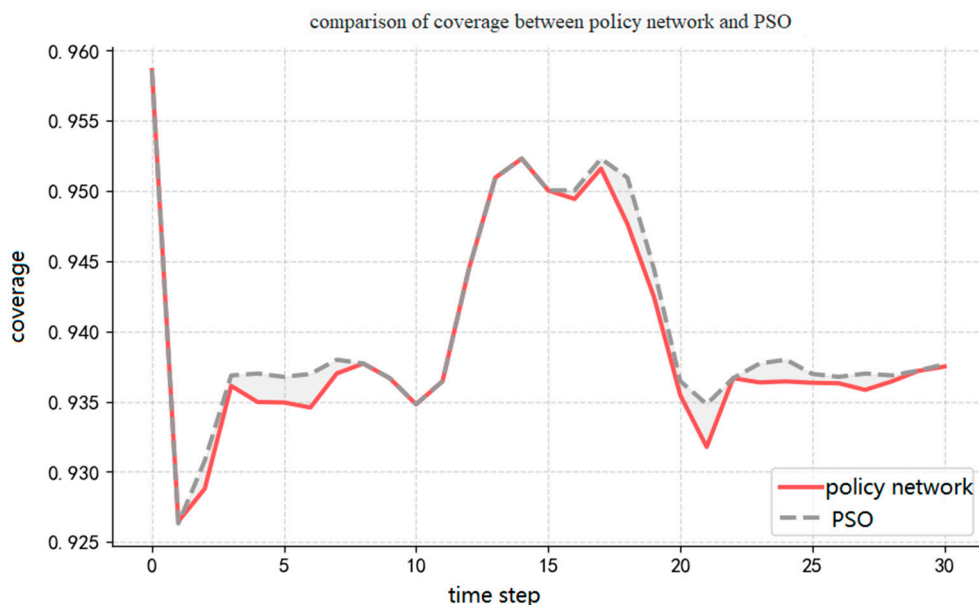


Figure 6. comparison of coverage between trajectory policy network and PSO.

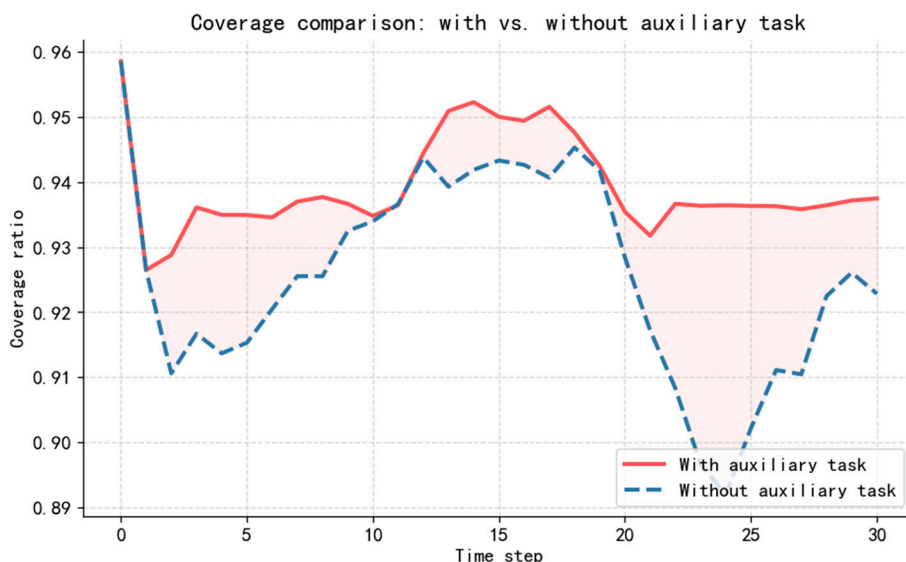


Figure 7. with & without auxiliary task comparison.

Figure 8 describes the experimental results of the trajectory policy network proposed in this paper and the particle swarm algorithm in terms of computational efficiency. The average computation time of the particle swarm algorithm is 449.19 s, while the trajectory policy network requires only 66.58 s. This result represents an improvement in computation speed of approximately 6.7 times that of the particle swarm algorithm.

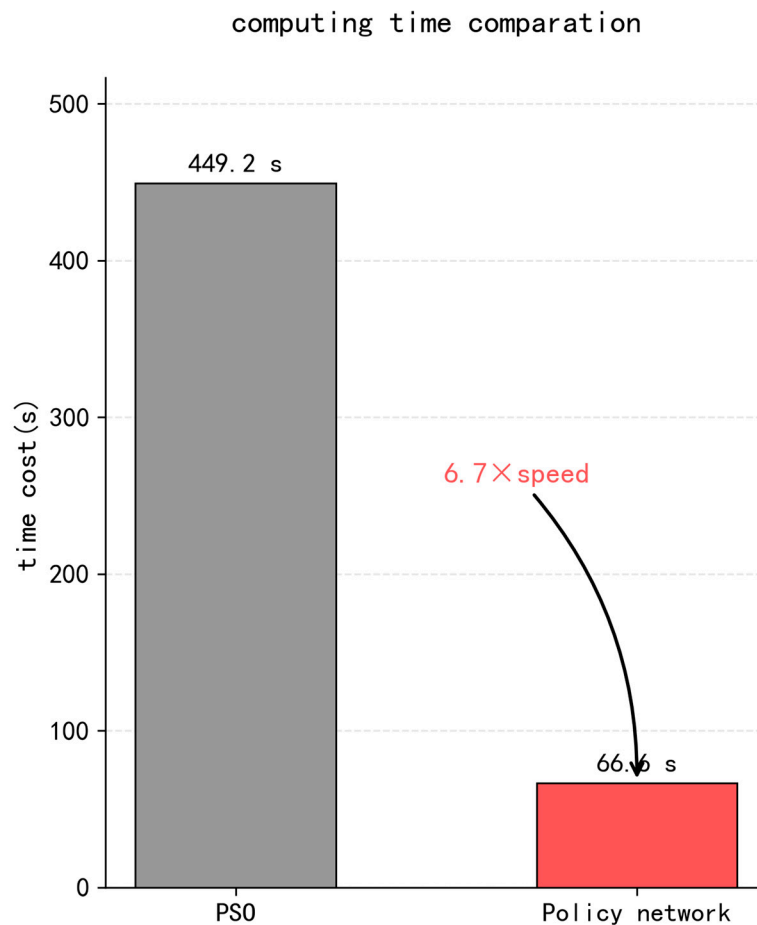


Figure 8. computing time comparison.

6. Discussion

This paper proposes an imitation learning-based trajectory planning method to solve the problem of insufficient real-time performance of multi-radar trajectory planning in complex electromagnetic jamming environments. This paper introduces an effective coverage rate metric for key areas based on detection probability. It constructs a performance evaluation model that comprehensively reflects the effects of radar deployment, trajectory maneuvering, and jamming. On this basis, a trajectory decision model targeting coverage rate improvement is designed. Expert demonstration data are generated using heuristic trajectory planning methods. The paper employed imitation learning to perform supervised training of the neural network, achieving effective approximation of the expert trajectory decision strategy.

Simulation experimental results demonstrate that the proposed method can significantly reduce the computational complexity of online trajectory decision-making while maintaining high coverage performance. It achieves fast and stable heading control output that satisfies the real-time trajectory planning requirements of multi-radar systems in complex electromagnetic environments. This method avoids the drawbacks of traditional methods such as high computational complexity and long computation time. It also avoids the problems of long training time and difficulty in convergence associated with training neural networks for decision-making from scratch. This method achieves a combination of the two approaches.

Author Contributions: Conceptualization, M.L. and J.G.; methodology, X.G. and M.L.; software, X.G.; validation, X.G.; formal analysis, X.G.; investigation, M.L. and J.G.; resources, K.G.; data curation, K.G.; writing—original

draft preparation, X.G.; writing—review and editing, M.L.; visualization, X.G.; supervision, M.L.; project administration, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by National Key R&D Program of China (2021YFA1000401), National Natural Science Foundation of China under Grant U22B2049 and Grant U23B2012.

Data Availability Statement: The data supporting the findings of this study are not publicly available due to confidentiality restrictions.

Acknowledgments: In addition, we would like to thank the anonymous reviewers who have helped to improve the paper.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
PSO	Particle Swarm Optimization
CFAR	Constant False Alarm Rate
SNR	Signal-to-Noise Ratio
SINR	Signal-to-Interference-plus-Noise Ratio
DQN	Deep Q-Network
DDPG	Deep Deterministic Policy Gradient
PPO	Proximal Policy Optimization
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
BC	Behavioral Cloning

References

1. LING H.; LI R.; BAI L.; HE W. Research on Russian Electronic Warfare Equipment and Its Application in the Russia–Ukraine Conflict. *Aero Weaponry* **2023**, *30*, 32–36.
2. Li, X.; Zhang, T.; Yi, W.; Kong, L.; Yang, X. Radar Selection Based on the Measurement Information and the Measurement Compensation for Target Tracking in Radar Network. *IEEE Sensors J.* **2019**, *19*, 7923–7935, doi:10.1109/JSEN.2019.2918866.
3. Tian, T.; Zhang, T.; Kong, L. Timeliness Constrained Task Scheduling for Multifunction Radar Network. *IEEE Sensors J.* **2019**, *19*, 525–534, doi:10.1109/JSEN.2018.2878795.
4. Jiang, W.; Qi, Z.; Ye, Z.; Wan, Y.; Li, L. Research on Cooperative Detection Technology of Networked Radar Based on Data Fusion. In Proceedings of the 2021 2nd China International SAR Symposium (CISS); IEEE: Shanghai, China, November 3 2021; pp. 1–5.
5. YAN J.; BAI G.; HUANG J.; DU L.; SONG T.; LIU H. Dynamic Coverage Track Optimization Method for Multi-Radar Cooperative Area. *Journal of Radars* **2023**, *12*, 541–549.
6. Lu, X.; Yi, W.; Kong, L. Joint Online Route Planning and Resource Optimization for Multitarget Tracking in Airborne Radar Systems. *IEEE Systems Journal* **2022**, *16*, 4198–4209, doi:10.1109/JSYST.2021.3116020.
7. Duan, H.; Zhao, J.; Deng, Y.; Shi, Y.; Ding, X. Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning. *IEEE Trans. Aerospace. Electron. Syst.* **2021**, *57*, 706–720, doi:10.1109/TAES.2020.3029624.
8. ZHANG X.; HU Y.; LI W.; ZHANG S.; PANG Q. An Improved Multi-UAV Coverage Path Planning Method. *Journal of Ordnance Equipment Engineering* **2020**, *41*, 215–221.
9. YANG, X.; WANG, R.; ZHANG, T. Intelligent Optimization Algorithms for UAV Swarm Path Planning: A Review. *Journal of Control Theory and Applications* **2020**, *37*, 2291–2302, doi:10.7641/CTA.2020.00157.
10. Zhang, H.; Cheng, H.; Wang, X.; Zhu, L.; Jiao, D.; Qiu, Z. Improved Secretary Bird Optimization Algorithm for UAV Path Planning. *Algorithms* **2026**, *19*, 64, doi:10.3390/a19010064.
11. Jiang, S.; Yu, T.; Cui, S. Improved Black-Winged Kite Algorithm and Its Application in Unmanned Aerial Vehicle Path Planning. *Journal of Supercomputing* **2026**, *82*, 99, doi:10.1007/s11227-025-08199-2.

12. Wei, J.; Zhu, W.; Xu, Q.; Li, Y.; Zhao, Y.; Li, Z.; Zhang, R.; Gu, Y.; Song, J.; Wang, Y.; et al. GeoSSA: Geometric Sparrow Search Algorithm for UAV Path Planning and Engineering Design Optimization 2026.
13. LIU R.; WANG B.; LI P. Joint Reconnaissance Mission Planning Based on PSO-SAS. *Telecommunication Engineering* **2020**, *60*, 1336–1341.
14. Besada-Portas, E.; De La Torre, L.; De La Cruz, J.M.; De Andrés-Toro, B. Evolutionary Trajectory Planner for Multiple UAVs in Realistic Scenarios. *IEEE Trans. Robot.* **2010**, *26*, 619–634, doi:10.1109/TRO.2010.2048610.
15. Shi, C.; Dai, X.; Wang, Y.; Zhou, J.; Salous, S. Joint Route Optimization and Multidimensional Resource Management Scheme for Airborne Radar Network in Target Tracking Application. *IEEE Systems Journal* **2022**, *16*, 6669–6680, doi:10.1109/JSYST.2021.3134657.
16. Zhang, Y.; Pan, M.; Han, Q.; Long, W.; Yang, S. Joint Power Allocation and Route Planning Scheme for Multitarget Tracking in Airborne Radar Network Under Multiuncertainty. *IEEE Sensors J.* **2023**, *23*, 7705–7718, doi:10.1109/JSEN.2023.3249782.
17. Yuan, M.; Lian, M.; Zhang, Y.; Li, C.; Shi, J.; Zhang, T. RadarImputationNet: An Enhanced Long-Term Diffusion Model for Automotive Radar Interference Suppression and Signal Recovery. *IEEE Sensors J.* **2025**, *25*, 24693–24711, doi:10.1109/JSEN.2025.3565263.
18. XIONG, S.; LI, Y.; OUYANG, Q.; WANG, Z. Review of UAV Swarm Trajectory Planning Based on Reinforcement Learning. *Space Electronic Technology* **2025**, *22*, 1-8+123, doi:10.3969/j.issn.1674-7135.2025.06.001.
19. Tang, J.; Yang, S.; Chen, S.; Li, Q.; Yin, Q.; Zhou, S. Research on UAV Autonomous Trajectory Planning Based on Prediction Information in Crowded Unknown Dynamic Environments. *Sensors* **2025**, *25*, 7343, doi:10.3390/s25237343.
20. Li, Y.; Li, G.; Wang, X. Research on Trajectory Planning of Autonomous Vehicles in Constrained Spaces. *Sensors* **2024**, *24*, 5746, doi:10.3390/s24175746.
21. XU Z.; CHEN M.; HAN Z.; SHAO S. Dynamic Route Planning for Low-Altitude Aircraft Based on TCP-DQN in Complex Environments. *Robot* **2025**, *47*, 383–393, doi:10.13973/j.cnki.robot.240341.
22. GAO J.; HU X.; JIANG Z. Improved DDPG Algorithm for UAV Path Planning. *Computer Engineering and Applications* **2022**, *58*, 264–272.
23. Xu, Y.; Wei, Y.; Wang, D.; Jiang, K.; Deng, H. Multi-UAV Path Planning in GPS and Communication Denial Environment. *Sensors* **2023**, *23*, 2997, doi:10.3390/s23062997.
24. WANG, Y.; GUAN, Z.; LI, Y. Warm Pursuit Decision-Making Based on Trajectory Prediction and Distributed MADDPG. *Journal of Computer Applications* **2024**, *44*, 3623–3628, doi:10.11772/j.issn.1001-9081.2023101538.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.