Imitation Learning of Jamming Strategies Under Low-Quality Expert Database Conditions

Tianjian Yang [1], Siyi Cheng [1], You Chen [1, a], Dejiang Lu [1], Xing Wang [1], Wei Liu [1], Haoyang Li [1], Xi Zhang [1]

[1] Aviation Engineering School, Air Force Engineering University, Xi'an, 710038, China.

[a] chenyoushky@163.com

Abstract. This paper proposes an improved generative adversarial imitation learning (GAIL) method for multi-functional radar (MFR) jamming decision-making, addressing the poor initial jamming effect of traditional deep reinforcement learning (DRL) and the high database quality requirement of basic GAIL. By introducing the behavior model (BM) and negative database (ND), the proposed method can improve the agent's imitation accuracy of the expert policy in low-quality data environments, avoiding learning ineffective jamming strategies. The agent strives to mimic the expert strategy during offline training through continuous updates. In online application, the trained Actor network directly informs jamming pattern decisions. The proposed method is validated through comparison with four other algorithms, including random strategy and basic GAIL. It effectively reduces MFR entry into high-threat states and improves aircraft survivability, offering a robust solution for enhancing aircraft survivability in complex electromagnetic environments.

Keywords: Jamming decision-making; Generative adversarial imitation learning; Behavioral imitation; Negative database.

1. Introduction

Multifunction Radar (MFR) is an advanced radar system capable of performing multiple tasks simultaneously, such as target detection, tracking, and identification[1], [2], [3], [4]. It is widely used in military and aviation fields. In the aircraft penetration scene, when the MFR is in a high - threat - level working state, the aircraft's survivability faces a severe challenge. To address this challenge, aircraft usually transmit jamming signals with different jamming styles to MFR. However, the effectiveness of different jamming styles varies significantly depending on the working state of the MFR. Therefore, reasonable jamming decision-making is of critical importance for enhancing the survivability of aircraft.

In the field of jamming decision-making, traditional heuristic algorithms have been widely applied, such as genetic algorithms [5], cuckoo search algorithms [6], artificial bee colony algorithms [7], and sparrow search algorithms [8]. With the advancement of computer technology, data-driven methods represented by deep reinforcement learning (DRL) have gained increasing attention due to their "model-free" advantages, offering new approaches to solving jamming decision-making problems. Considering scenarios where MFR working modes dynamically change based on mission requirements, Zhang et al. [9] proposed an Exploratory Deep Deterministic Policy Gradient algorithm to achieve dynamic adjustment of multi-step jamming power. Pan et al. [10] utilized the HPPO algorithm to intelligently select the jamming style and power of the jammer, achieving superior jamming effectiveness. Wang et al. [11] achieved joint optimization of hybrid discrete (jamming tasks) and continuous (jamming power) control variables and extended its application to networked radar systems.

However, as the electromagnetic environment becomes increasingly complex, the competition between radar and electronic countermeasures grows more intense [12]. Conventional DRL suffers from limitations, including dependence on manually defined reward functions, high subjectivity, and slow exploration processes caused by expansive action spaces [13]. Traditional DRL relies on continuous interaction with the environment to train agents. However, this process does not guarantee the learning of effective strategies. Even if good strategies are eventually learned, the

time-consuming nature of this approach means that the aircraft's initial jamming strategy will likely be suboptimal, posing a significant risk to its survival. In 2016, Jonathan et al. introduced Generative Adversarial Imitation Learning (GAIL) [14], circumventing the challenges associated with reward function design. Yang et al. [15] applied GAIL to jamming strategy learning, allowing aircraft to adopt effective strategies early in the jamming phase by combining offline learning and online implementation. Nevertheless, the learning database is too idealistic, and the databases available in practical scenarios are typically of poor quality, greatly hindering the agent's learning effectiveness.

In summary, this paper proposes a jamming strategy learning method based on improved GAIL under low-quality database conditions. By combining Generative Adversarial Networks (GAN) [16] with DRL, the agent can imitate expert databases without relying on reward functions. On this basis, a behavior model (BM) is introduced to extract useful information even from low-quality databases. Additionally, a negative database (ND) is constructed to prevent the agent's strategy from learning the low-quality portions of the expert database. The effectiveness of the proposed algorithm is validated through simulation experiments.

2. Radar Countermeasure Scenario

When the MFR radar does not detect a target, it continuously emits radar signals into the air to perform search tasks. These signals propagate through space, and once an aircraft enters the radar's detection range, the radar receiver will capture the reflected echoes with sufficient energy. Under no-jamming conditions, if the signal-to-noise ratio of the echo exceeds the preset threshold, the MFR will transition from the initial search state to the confirmation state [17].

In the confirmation state, the radar performs further detection and analysis on the target to obtain more precise information, such as position and velocity. Once the target is successfully confirmed, the MFR will automatically switch to the tracking state to enable continuous monitoring and identification of the target. In the tracking state, the radar prioritizes stable tracking and identification of the target to ensure continuous monitoring.

Nevertheless, to improve their survival capabilities, aircraft implement appropriate jamming strategies tailored to the radar's varying working states. By emitting different types of jamming signals, aircraft can effectively prevent the escalation of the radar's threat level, thereby reducing the risk of being locked onto and attacked by the radar. The radar countermeasure scenario is illustrated in Figure 1.

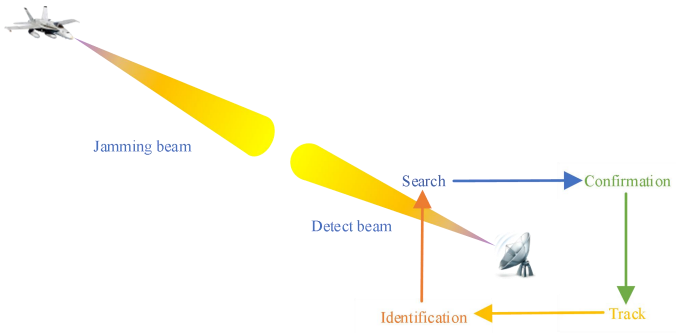

Fig. 1 Radar countermeasure scenario

3. Problem Formulation

3.1 Problem Formulation.

In the adversarial game considered in this paper, the MFR and the aircraft are treated as the environment and the agent, respectively. During the jamming decision-making process, the aircraft selects actions based on the current state, and state transitions depend solely on the current state and action, aligning with the characteristics of a Markov Decision Process (MDP). Therefore, modeling

the jamming decision-making process using MDP enables more effective strategy optimization [18]. Based on the principles of MFR working states, the MFR state transition model is illustrated in Figure 2 [15].

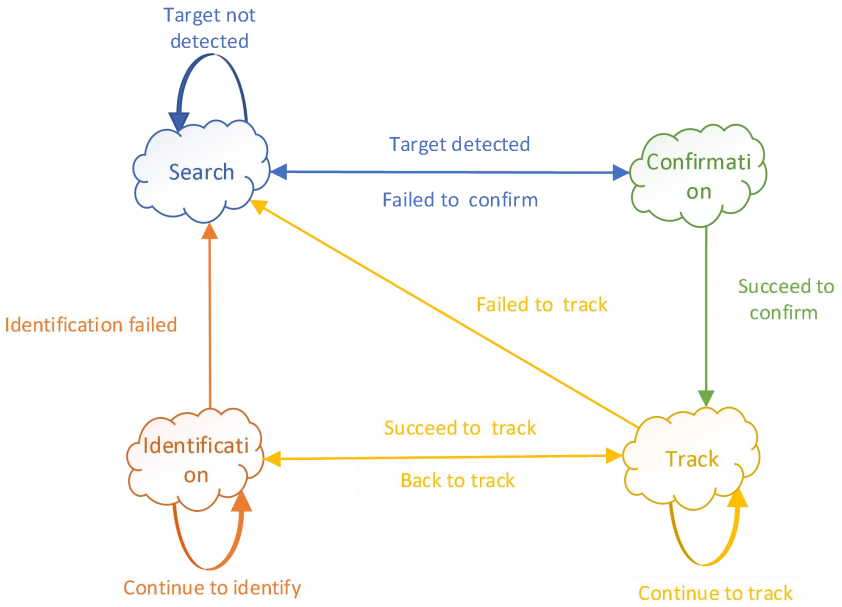

Fig. 2 Radar state transition model

MDP provides a crucial foundation for the development of reinforcement learning. The essential components of the MDP model are the state space S, action space A, reward function R, and state transition probability P [11]. Here, $P(s_{t+1} \mid s_t, a_t)$ represents the probability of the aircraft's state transitioning from s_t to s_{t+1} when the jamming action a_t is taken, with its value ranging between [0, 1]. $R(s_t, a_t, s_{t+1})$ represents the reward obtained by the aircraft after completing the corresponding state transition. γ denotes the discount factor.

During the jamming style selection task, the aircraft chooses suitable jamming styles according to the radar's working state, aiming to either lower the radar's threat level or maintain it at a low threat level. To scientifically and quantitatively evaluate the effectiveness of the aircraft's jamming style selection, a reward function is specifically designed, which uses the cumulative reward value obtained by the aircraft within a single episode as the evaluation metric, as detailed below:

$$E = \sum_{t}^{STEP} r_t \tag{1}$$

Where, T represents the total time steps in the episode, and r_t denotes the immediate reward value at time step t.

3.2 Action Space.

In the jamming task scenario studied in this paper, the aircraft needs to select an appropriate jamming style. Given this, when constructing a DRL model to optimize the aircraft's jamming action, the decision variable in the action space is set as the jamming style selected by the aircraft, as specifically expressed below:

$$a_t = J_t \tag{2}$$

Where, J_t represents the jamming style selected at time step t. Since the aircraft can only support a limited number of jamming styles, the range of action choices is explicitly constrained:

$$0 < a_t \le Q \tag{3}$$

Where, Q denotes the maximum number of jamming styles the aircraft is capable of emitting.

3.3 State Space.

When constructing the aircraft jamming style selection model under the DRL framework, the selection of state variables is crucial for accurately characterizing the environmental state of the aircraft. The aircraft's rewards are closely tied to changes in the radar's working state, which are significantly influenced by the different jamming styles used by the aircraft. Therefore, this paper selects the following variables as the aircraft's state: the current working state of the MFR W_t, the previous working state W_{t-1}, and the action executed by the aircraft at the previous time step a_{t-1}.

$$s_t = (W_t, W_{t-1}, a_{t-1}) \tag{4}$$

This paper categorizes the radar's working states into four types: search, confirmation (Com.), tracking, and identification (Iden.). Under this classification framework, there are constraints on the radar's working states:

$$0 \le W_t, W_{t-1} \le 3 \tag{5}$$

3.4 Reward Function Design.

The reward function design, which evaluates the goodness of the agent's actions, is crucial for algorithm convergence. Nevertheless, accurately modeling the reward function is confronted with high complexity and difficulty in parameter adjustment. Additionally, as environmental uncertainty increases, traditional reward functions find it hard to objectively and accurately assess the agent's actions. In the GAIL process, the reward function is designed by comparing the similarity between the agent's policy and the expert's policy to guide the agent's learning, which is specifically implemented by constructing a discriminator.

On one hand, the discriminator takes the current state-action pair as input and outputs a probability value. This value indicates the likelihood that the current state-action pair belongs to the generated strategy. It intuitively reflects the difference between the generated strategy and the expert strategy. This evaluative feedback can replace traditional manually designed reward functions, guiding the generator to update its parameters. This significantly reduces modeling difficulty and parameter sensitivity.

On the other hand, constructing a discriminator allows for generating specific rewards tailored to the current strategy. This "custom-made" reward mechanism reduces the subjective bias and empirical limitations of manually setting reward functions. It also automatically adjusts reward signals based on dynamic changes in the environment. This enhances the algorithm's generalization capability and adaptability. Compared to traditional reinforcement learning methods, this approach avoids biases and shortcomings from manually designed reward functions. It is advantageous in dynamic radar countermeasure scenarios. The specific configuration of the reward function is detailed in Section 4.3 of this paper.

4. GAIL Framework for Jamming Decision-making

GAIL is an imitation learning method based on DRL and GAN. Building on this, this paper establishes a GAIL-based jamming decision-making framework, as shown in Figure 3. The basic GAIL framework typically consists of three stages: expert database construction, imitation learning, and online application [15]. In the expert memory library construction stage, expert strategies are generated based on the experience of pilots or domain experts and stored in the expert database. During the imitation learning phase, the Actor-Critic network of the PPO algorithm[19] serves as the generator to develop jamming decision strategies tailored to the jamming confrontation scenario. Additionally, a Discriminator is introduced to differentiate between expert strategies and generated strategies, with the generator and discriminator being updated inversely according to the discrimination outcomes. The generator and discriminator reach a Nash equilibrium through mutual competition, at which point the agent's strategy approximates the expert strategy. During the online

application phase, various states are fed into the agent to determine the appropriate actions for those states.

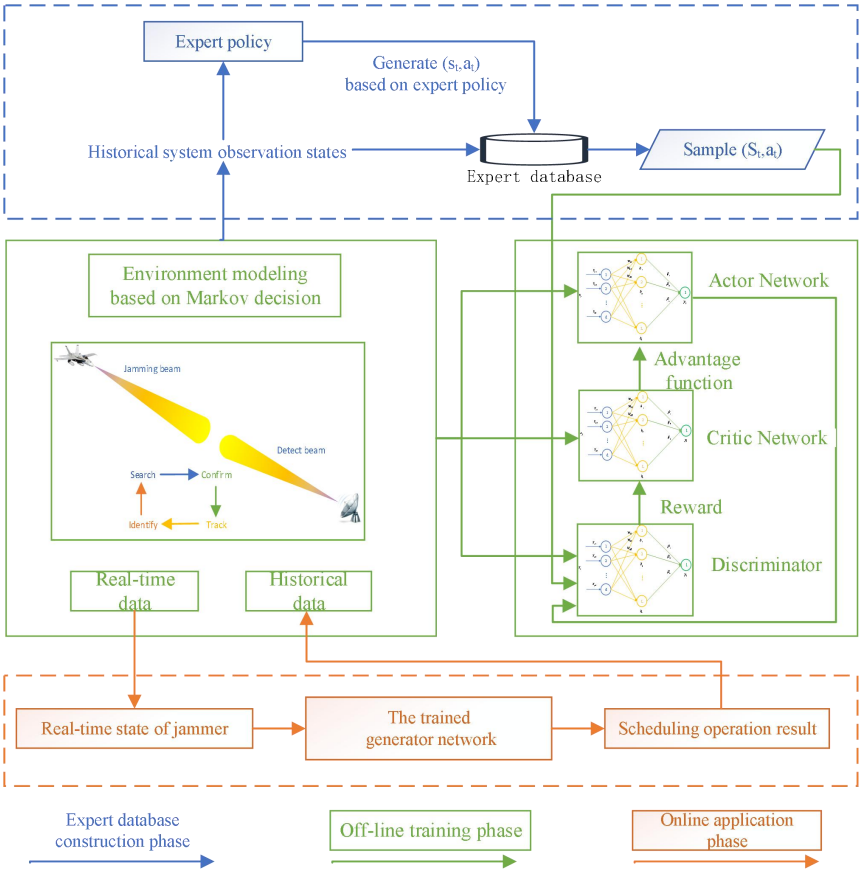

Fig. 3 GAIL framework for jamming decision-making

4.1 Constructing The Expert Database.

During flight, the aircraft selects different jamming styles to jam with the MFR, causing changes in the MFR's working state. As shown in Figure 2, the MFR's working state can rise, fall, or remain unchanged. Jamming actions that cause the MFR's working state to decrease or remain at a low threat level can be considered effective. The state-action pairs corresponding to effective jamming are stored to construct the expert database.

4.2 Constructing The Generator Network Based on The PPO Algorithm.

The generator network in this paper is based on the Actor-Critic architecture [20]. The Actor network learns the jamming strategy by observing the aircraft state in real time and using the policy gradient method. The Critic network uses data from the interaction between the Actor and the environment to learn the state value function, which provides an evaluation basis for policy updates. To enhance training stability, the network employs the PPO algorithm proposed by OpenAI in 2017 [21]. This algorithm introduces a constraint mechanism on the policy update magnitude, effectively preventing the new policy from deviating too far from the current policy. It retains the advantages of the policy gradient method while significantly improving training stability and sample efficiency. The principle of the PPO algorithm is as follows:

The optimization objective function of PPO is as follows:

$$L^{CLIP}(\theta) = E_t\left[\min\left(r_t(\theta)A_t, \text{clip}\left(r_t(\theta), 1-\varepsilon, 1+\varepsilon\right)A_t\right)\right] \tag{6}$$

Where:

- $r_t(\theta)$ is the probability ratio, representing the ratio of the probabilities of selecting an action under the new policy and the old policy in the same state.

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \tag{7}$$

Where, $\pi_\theta(a_t \mid s_t)$ denotes the probability of the new policy choosing action a_t in state s_t, whereas $\pi_{\theta_{\text{old}}}(a_t \mid s_t)$ is the probability of the old policy under the same state.

- A_t is the advantage function, and its calculation formula is as follows:

$$A_t = R_t + \gamma V(s_{t+1};\omega) - V(s_t;\omega) \tag{8}$$

Where, $V(s_t;\omega)$ represents the state value output by the Critic network when the input is s_t, indicating the expected return the agent can obtain by following the current policy π; R_t is the reward value at time t.

- The function of $\text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)$ is to confine $r_t(\theta)$ within the range $[1-\varepsilon, 1+\varepsilon]$, avoiding significant policy variations. In this study, ε is assigned a value of 0.1.

The pseudocode for the PPO algorithm is as follows:

Algorithm 1 PPO Training

Initialize the parameters θ of the policy network π.
Initialize the parameters of the value function network V.
While $t < T$:
  Collect a set of samples D by interacting with the environment using policy π.
  Calculate the advantage function A_t and the cumulative reward R_t for each sample.
  While $k < K$:
    Randomly sample a batch of data from the dataset D.
    Randomly sample a batch of data from the dataset D.
    Calculate the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$
    Calculate the clipped probability ratio $\text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)$
    Calculate the policy loss $L^{CLIP}(\theta)$.
    Calculate the value function loss $L^{VF}(\varphi)$.
    Calculate the total loss $L(\theta,\phi) = L^{\text{CLIP}}(\theta) - c_1 \cdot L^{\text{VF}}(\phi) + c_2 \cdot S[\pi_\theta](s_t)$
    Update the parameters θ and φ using gradient descent.
  $\theta_{old} \leftarrow \theta$

The generator network based on the PPO algorithm can prevent excessive policy updates through the clipping mechanism, ensuring the stability of the training process. Additionally, PPO uses sampled data for multiple training passes, enhancing sample efficiency, reducing sample complexity, and enabling effective learning from the expert database.

4.3 Constructing The Discriminator Network.

The discriminator network evaluates how closely the generator mimics the expert strategy. When the state-action pair (s_t, a_t) generated by the generator is input into the discriminator network, it outputs a real number in the range [0,1], where $D(s_t, a_t)$ represents the probability that the input state-action pair comes from the generator strategy. The output of the generator guides the updates of the generator network, causing the generator strategy to increasingly approximate the expert strategy, allowing the state-action pairs generated by the generator to be misclassified as those produced by the expert strategy. Therefore, the immediate reward R_t at time t is set as:

$$R_t = -\log D_\phi(s_t, a_t)_{\pi_G} \tag{9}$$

Where, $D_\phi(s_t, a_t)$ represents the discriminator's judgment result; π_G denotes that the state-action pair (s_t, a_t) comes from the generator strategy; ϕ represents the parameters of the discriminator network.

The discriminator and generator compete against each other, with the discriminator continuously updating to improve its ability to distinguish between the generator strategy and the expert strategy. Therefore, the loss function F of the generator is defined as:

$$F(\phi) = -E[\log D_\phi(s_t, a_t)_{\pi_G}] - E[1 - \log D_\phi(s_t, a_t)_{\pi_E}] \tag{10}$$

Where, $E[\bullet]$ represents the expected value; π_E denotes the expert strategy.

5. Improvement Strategy

The transition probabilities of the MFR's working states follow a Markov model. When the aircraft selects a jamming style that does not match the radar's working state, the MFR's working state may still decrease or remain at a low threat level. This information is stored in the expert database. However, this implies that the expert database contains low-quality strategies, which can negatively impact the learning performance of the agent. Therefore, this paper proposes an improved strategy for generative adversarial imitation learning.

5.1 Behavior Model.

The BM is a model used to represent the distribution of behaviors in expert data. In this paper, the BM is employed to learn behavioral strategies from expert data, enabling the extraction of useful information even from low-quality datasets. The BM learns by maximizing the log-likelihood of the observed behaviors, as shown in the formula:

$$\theta_{\text{bm}} = \arg\max_{\theta_{\text{bm}}} \llcorner_{\tau \sim D_\mu} \left[\sum_{t=1}^{|T|} \log \pi_{\theta_{\text{bm}}}(a_t \mid s_t) \right] \tag{11}$$

Where, D_μ is the expert dataset, $\pi_{\theta_{\text{bm}}}$ is the strategy of the behavior model, θ_{bm} represents the parameters of the BM, and T is the maximum number of steps. By employing this method, the BM can identify the action distribution within the expert data, laying the groundwork for future strategy optimization.

After obtaining the behavior distribution from the expert database using the behavior model, the Advantage-Weighted Behavior Model (ABM) is employed to further optimize the behavior model [22]. ABM introduces the advantage function to weight the BM, making the strategy more inclined to select actions that are likely to succeed in the current task. Specifically, the objective function of ABM is as follows:

$$\theta_{\text{abm}} = \arg\max_{\theta_{\text{abm}}} \llcorner_{\tau \sim D_\mu} \left[\sum_{t=1}^{|T|} \log \pi_{\theta_{\text{abm}}}(a_t \mid s_t) f\left(R(\tau_{t:N}) - \hat{V}_{\pi_i}(s_t) \right) \right] \tag{12}$$

Where, $R(\tau_{t:N})$ is the cumulative reward from time step t to N, $\hat{V}_{\pi_i}(s_t)$ is the value function of the policy π_i at state s_t, and f is a non-negative increasing function used to weight the advantage.

5.2 Constructing The Negative Database.

As shown in Section 4.1, the traditional method of constructing the expert database only stores state-action pairs with good jamming effects, while discarding those with poor jamming effects. However, this approach has certain limitations. Inspired by the TOPSIS algorithm [23], [24], a good strategy should be as close as possible to the expert strategy while being distinctly different from poor strategies. Therefore, this paper chooses to collect state-action pairs corresponding to both

good and poor jamming effects, constructing a positive database and an ND, respectively. Extracting strategies from these two databases will result in two distinct BMs.

Combining the two BMs, the advantage-weighted model, and the ND, the discriminator's immediate reward R_t at time t is updated as:

$$R_t = A \bullet \pi_{\text{bm1}}(a \mid s) - A \bullet \pi_{\text{bm2}}(a \mid s) \tag{13}$$

Where, $A = -\log D_\phi(s_t, a_t)_{\pi_G}$ represents the advantage function, $\pi_{\text{bm1}}(a \mid s)$ and $\pi_{\text{bm2}}(a \mid s)$ denote the probabilities of taking action a in state s under the first and second behavior models, respectively.

Through this improvement to the generative adversarial imitation learning algorithm, the agent can efficiently learn the expert strategy and simultaneously avoid the suboptimal parts of the expert database, resulting in a favorable learning performance.

6. Simulation Experiments

6.1 Simulation Scenario Construction and Basic Data Settings.

In the radar confrontation scenario of this paper, the two parties involved in the game are an aircraft and an MFR, both operating in the same frequency band by default. The aircraft can intelligently select jamming styles, making flexible decisions based on environmental changes. There are three optional jamming styles, labeled as Jamming Style 1, Jamming Style 2, and Jamming Style 3 [15]. When the aircraft jams the MFR, the radar's working state changes follow a Markov process, with the state transition probabilities shown in Table 1. After constructing the scenario and related data, we further elaborate on the parameters of generative adversarial imitation learning, as shown in Table 2.

Table 1. Working state transition probability table

Current Mode	Jamming Type1				Jamming Type2				Jamming Type3			
	Search	Con.	Track	Iden.	Search	Con.	Track	Iden.	Search	Con.	Track	Iden.
Search	0.9	0.1	0	0	0.3	0.7	0	0	0.6	0.4	0	0
Con.	0.8	0	0.2	0	0.5	0	0.5	0	0.2	0	0.8	0
Track	0.2	0	0.7	0.1	0.9	0	0.09	0.01	0.11	0	0.8	0.09
Iden.	0.15	0	0.1	0.75	0.7	0	0.19	0.11	0.12	0	0.03	0.85

Table 2. Parameters of neural network

Parameters of PPO	Value	Parameters of PPO	Value
Actor network		Critic network	
Input layer	3	Input layer	3
Hidden layer1	128	Hidden layer1	128
Hidden layer2	64	Hidden layer2	64
Output layer	3	Output layer	1
Discount factor	0.999	Discount factor	0.999
Learning rate	1e-5	Learning rate	1e-5
Min batch size	200	Optimizer	Adam
Parameters of GAIL	**Value**	**Parameters of GAIL**	**Value**
Discriminator network		Discriminator network	
Input layer	6	Discount factor	0.999
Hidden layer1	64	Learning rate	1e-4
Hidden layer2	32	Min batch size	200
Output layer	1	Optimizer	RMSprop
Parameters of BM			
Input layer	3	Discount factor	0.999
Hidden layer1	128	Learning rate	1e-4

Hidden layer2	64	Min batch size	200
Output layer	3	Optimizer	Adam

6.2 Offline Training Results Analysis.

In the offline training stage, the agent iteratively updates itself to mimic the expert's strategy. To validate the effectiveness of the proposed GAIL combined with the BM and ND, it is compared with four other algorithms. These four algorithms are the random strategy (Random), the basic GAIL (GAIL), the GAIL algorithm combined with the BM (GAIL_BM), and the GAIL algorithm combined with the ND (GAIL_ND). To effectively compare the performance of the five methods, this study defines a reward function, as presented in Table 3. This reward function is used exclusively for comparison and is not involved in the agent's learning process. Set the number of steps in each episode to 200 and the total number of episodes T to 2000. The effectiveness comparison of the algorithms is illustrated in Figure 4.

Table 3. The corresponding reward of radar working state transition

Current Mode	Next Mode			
	Search	Con.	Track	Iden.
Search	2	-2	NAN	NAN
Con.	3	NAN	-4	NAN
Track	4	NAN	-8	-10
Iden.	5	NAN	0	-10

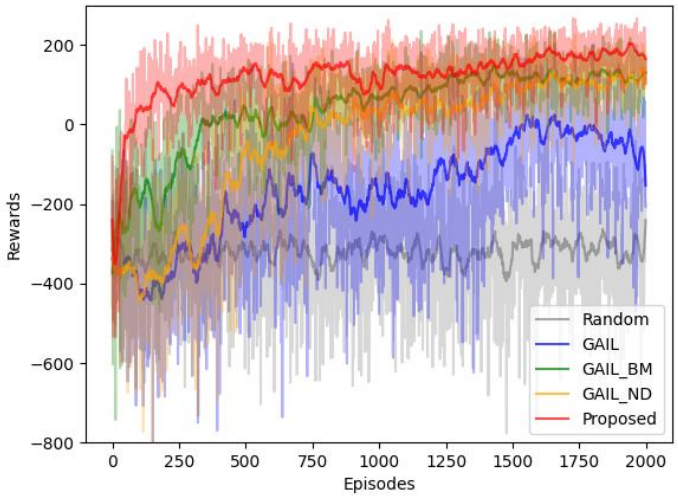

Fig. 4 The effectiveness comparison of five strategies

As illustrated in Figure 4, the GAIL algorithm is capable of mimicking the expert database, but the imitation performance is relatively poor, and the learning process exhibits instability with considerable fluctuations. By combining the BM and the ND with the GAIL algorithm, the learning effectiveness is improved, and the learning process becomes more stable. Specifically, the introduction of the behavior model enables the agent to better extract useful information from low-quality data, improving the accuracy of imitating the expert strategy; meanwhile, the construction of the negative database prevents the agent from learning low-quality jamming strategies, further enhancing the quality of the strategy. The combined effect of these two improvement strategies ensures that the improved GAIL algorithm outperforms the traditional GAIL algorithm in both learning efficiency and stability.

6.3 Online Application Results Analysis.

After completing the offline training phase, the Actor network trained by the GAIL algorithm can be directly applied to online jamming style decision-making. To validate the effectiveness of the proposed method, we conducted a comparative analysis of four jamming strategies based on the GAIL algorithm. Using each algorithm, we trained for 1000 episodes, resulting in four different agents, and conducted jamming decisions based on these agents.

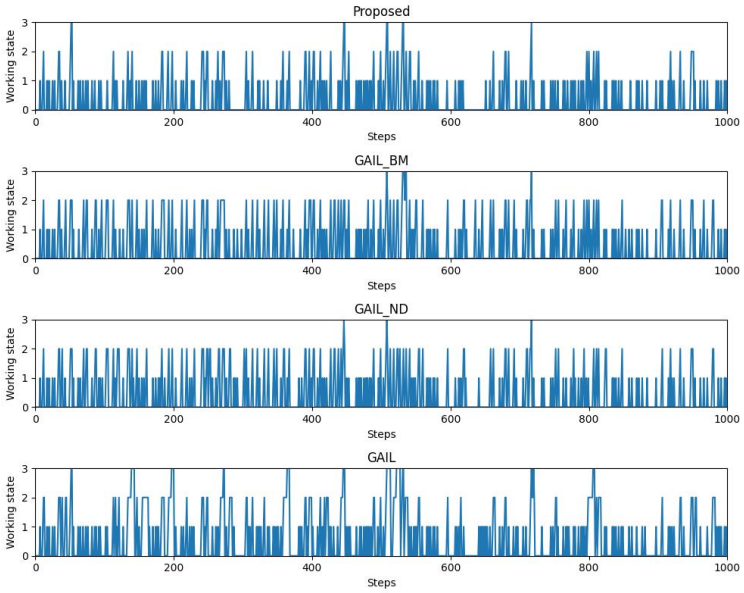

Fig. 5 Radar working states transition

As shown in the figure, the improved GAIL algorithm demonstrates a significant enhancement in online application effectiveness compared to the basic GAIL algorithm, with fewer instances of the MFR entering the identification state, effectively reducing the survival threat faced by the aircraft. Compared to the three improved methods, the jamming strategy implemented based on the method proposed in this paper allows the MFR to remain in the search state for the longest duration. Compared to the three improved methods, the jamming strategy implemented based on the method proposed in this paper allows the MFR to remain in the search state for the longest duration.

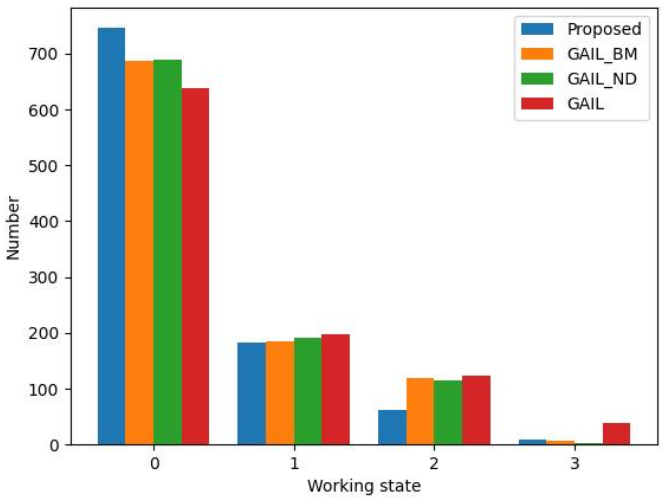

Fig. 6 Comparison of radar working state statistics

Table 4. Radar working state statistics

Working state	Proposed	GAIL_BI	GAIL_ND	GAIL
Search	746	688	690	639
Con.	183	185	182	198
Track	62	120	115	124
Iden.	9	7	3	39

As shown in Figure 6 and Table 4, when implementing jamming based on the GAIL algorithm, the radar enters the high-threat-level tracking and identification states 124 and 39 times, respectively. After improving the GAIL algorithm, the number of times the radar enters the tracking and identification states is significantly reduced. Among them, the method proposed in this paper achieves the best jamming effect, reducing the number of times the radar enters the tracking and identification states to 62 and 9, respectively, effectively mitigating the survival threat faced by the aircraft.

7. Conclusion

Traditional DRL methods face challenges in designing reward functions and require a long adaptation time to the environment when addressing jamming style selection problems. Using generative adversarial imitation learning algorithms enables the aircraft to adaptively select jamming styles from the beginning of the confrontation, but it imposes higher requirements on the expert database. This paper conducts jamming strategy learning under low-quality database conditions and draws the following conclusions:

(1) By adopting generative adversarial networks, the modeling complexity and parameter sensitivity of the reward function can be effectively reduced, addressing the difficulty in designing reward functions for complex jamming style selection problems.

(2) Introducing the BM and ABM can effectively enhance the agent's imitation of expert strategies.

(3) Constructing an ND can effectively prevent the agent from learning poor jamming strategies, thereby improving the agent's learning performance.

This paper improves generative adversarial imitation learning, enabling effective learning from low-quality databases. Future research could focus on more dynamic environments to enhance the applicability of the algorithm.

References

[1] Z. Zhang, X. Shi, and F. Zhou, “An Incremental Recognition Method for MFR Working Modes Based on Deep Feature Extension in Dynamic Observation Scenarios,” *IEEE Sensors Journal*, vol. 23, no. 18, pp. 21574–21587, Sep. 2023, doi: 10.1109/JSEN.2023.3303023.

[2] N. Visnevski, V. Krishnamurthy, A. Wang, and S. Haykin, “Syntactic Modeling and Signal Processing of Multifunction Radars: A Stochastic Context-Free Grammar Approach,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 1000–1025, 2007, doi: 10.1109/JPROC.2007.893252.

[3] H. S. Mir and F. Ben Abdelaziz, “Cyclic Task Scheduling for Multifunction Radar,” *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 529–537, Jul. 2012, doi: 10.1109/TASE.2012.2197857.

[4] J. Ou, Y. Chen, F. Zhao, J. Liu, and S. Xiao, “Novel Approach for the Recognition and Prediction of Multi-Function Radar Behaviours Based on Predictive State Representations,” *Sensors*, vol. 17, no. 3, p. 632, Mar. 2017, doi: 10.3390/s17030632.

[5] H. Jiang, Y. Zhang, and H. Xu, “Optimal allocation of cooperative jamming resource based on hybrid quantum-behaved particle swarm optimisation and genetic algorithm,” *IET Radar Sonar Navig.*, vol. 11, no. 1, pp. 185–192, Jan. 2017, doi: 10.1049/iet-rsn.2016.0119.

[6] D. Lu, X. Wang, X. Wu, and Y. Chen, "Adaptive allocation strategy for cooperatively jamming netted radar system based on improved cuckoo search algorithm," *Def. Technol.*, vol. 24, pp. 285–297, Jun. 2023, doi: 10.1016/j.dt.2022.04.013.
[7] H. Xing, Q. Xing, and K. Wang, "A Joint Allocation Method of Multi-Jammer Cooperative Jamming Resources Based on Suppression Effectiveness," *Mathematics*, vol. 11, no. 4, p. 826, Feb. 2023, doi: 10.3390/math11040826.
[8] T. Yang, X. Wang, S. Cheng, Y. Chen, and X. Zhang, "Research on adaptive scheduling strategy of cooperative jamming resources for the anti-netted radar system," *AIP Advances*, vol. 14, no. 12, p. 125211, Dec. 2024, doi: 10.1063/5.0237486.
[9] Y. Zhang, W. Huo, C. Zhang, J. Pei, Y. Zhang, and Y. Huang, "Smart Noise Jamming Power Adjustment Using Exploratory Deep Deterministic Policy Gradient," in *2023 IEEE Radar Conference (RadarConf23)*, May 2023, pp. 1–6. doi: 10.1109/RadarConf2351548.2023.10149662.
[10] Z. Pan, Y. Li, S. Wang, and Y. Li, "Joint Optimization of Jamming Type Selection and Power Control for Countering Multifunction Radar Based on Deep Reinforcement Learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 4, pp. 4651–4665, Aug. 2023, doi: 10.1109/TAES.2023.3272307.
[11] Y. Wang, Y. Liang, and Z. Wang, "Hierarchical Reinforcement Learning-Based Joint Allocation of Jamming Task and Power for Countering Networked Radar," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–19, 2024, doi: 10.1109/TAES.2024.3467041.
[12] W. Zhang, D. Ma, Z. Zhao, and F. Liu, "Design of Cognitive Jamming Decision-Making System Against MFR Based on Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10048–10062, Aug. 2023, doi: 10.1109/TVT.2023.3261318.
[13] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation Learning: A Survey of Learning Methods," *ACM Comput. Surv.*, vol. 50, no. 2, p. 21, Jun. 2017, doi: 10.1145/3054912.
[14] J. Ho and S. Ermon, "Generative adversarial imitation learning," Curran Associates, Inc., 2016, pp.4565-4573.
[15] T. Yang, Y. Chen, S. Cheng, X. Wang, and X. Zhang, "Optimization of Jamming Type Selection for Countering Multifunction Radar Based on Generative Adversarial Imitation Learning," *IEEE Access*, vol. 13, pp. 17110–17119, 2025, doi: 10.1109/ACCESS.2025.3531016.
[16] J. Cheng, Y. Yang, X. Tang, N. Xiong, Y. Zhang, and F. Lei, "Generative Adversarial Networks: A Literature Review," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 12, pp. 4625–4647, Dec. 2020, doi: 10.3837/tiis.2020.12.001.
[17] Z. Xu *et al.*, "Adaptive Multi-Function Radar Temporal Behavior Analysis," *Remote Sensing*, vol. 16, no. 22, Art. no. 22, Jan. 2024, doi: 10.3390/rs16224131.
[18] L. Han, Q. Ning, B. Chen, Y. Lei, and X. Zhou, "Ground threat evaluation and jamming allocation model with Markov chain for aircraft," *IET Radar Sonar Navig.*, vol. 14, no. 7, pp. 1039–1045, Jul. 2020, doi: 10.1049/iet-rsn.2019.0433.
[19] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games," presented at the Neural Information Processing Systems, Mar. 2021. doi: 10.48550/arXiv.2103.01955.
[20] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid Actor-Critic Reinforcement Learning in Parameterized Action Space," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 2279–2285. doi: 10.24963/ijcai.2019/316.
[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," Aug. 28, 2017, *arXiv*: arXiv:1707.06347. doi: 10.48550/arXiv.1707.06347.
[22] N. Y. Siegel *et al.*, "Keep Doing What Worked: Behavioral Modelling Priors for Offline Reinforcement Learning," Jun. 17, 2020, *arXiv*: arXiv:2002.08396. doi: 10.48550/arXiv.2002.08396.
[23] X. Chen, X. Wang, H. Zhang, Y. Xu, Y. Chen, and X. Wu, "Interval TOPSIS with a novel interval number comprehensive weight for threat evaluation on uncertain information," *J. Intell. Fuzzy Syst.*, vol. 42, no. 4, pp. 4241–4257, 2022, doi: 10.3233/JIFS-210945.
[24] Y. Yin, R. Zhang, and Q. Su, "Threat assessment of aerial targets based on improved GRA-TOPSIS method and three-way decisions," *Math. Biosci. Eng.*, vol. 20, no. 7, pp. 13250–13266, 2023, doi: 10.3934/mbe.2023591.