Decision Modeling for Unmanned Swarm Suppression of Enemy Air Defenses Based on Deep Reinforcement Learning

Xiao Hu, Yonglin Lei, Fusong Luo, Hongfei Shi, Jiajun Zhu
College of Systems Engineering, National University of Defense Technology
Changsha 410003, China
email: huxiao14@nudt.edu.cn

***Abstract*—The Suppression of Enemy Air Defense (SEAD) mission is a critical component of Unmanned Aerial Vehicle (UAV) swarm operations, presenting a complex challenge for modeling and simulation. Machine Learning (ML), particularly Deep Reinforcement Learning (DRL), offers a promising approach to enhance UAV swarm SEAD effectiveness through intelligent decision-making. This paper, therefore, explores a modeling and simulation approach to intelligent combat equipment decision-making based on deep DRL. We establish a DRL modeling framework grounded in combat simulation and specifically construct an intelligent decision-making framework for UAV Swarm SEAD. Focusing on the attack decision-making problem, we present a case study utilizing the Dueling Deep Q-Network (Dueling DQN) algorithm for intelligent combat decision modeling. Preliminary experimental results demonstrate that the ML-based intelligent decision-making model achieves superior combat effectiveness compared to traditional knowledge engineering-based models.**

Keywords- UAV swarm ;SEAD; decision-making modeling; combat simulation ; Dueling DQN.

I. INTRODUCTION

Traditional manned aircraft assault methods face significant challenges in ensuring the safety of personnel and platforms against the modern air defense system. With the rapid development of UAV technology, employing UAV swarms is poised to become the predominant approach for executing SEAD tasks in the future [1].

The core challenge in achieving autonomous mission execution for UAV swarms lies in solving the problem of intelligent combat decision-making for their operations. Conventional UAV swarm combat decision-making primarily relies on knowledge engineering techniques, such as production rules and expert database systems [2]. However, these methods exhibit limitations, including difficulty in enumerating the complexity of the situational space, challenges in handling the inherent uncertainties of complex scenarios, and a lack of adaptive evolution in combat decision algorithms. Concurrently, the increasing credibility of unmanned combat simulation systems enables the generation of vast amounts of offensive and defensive combat data. This data can not only be used to evaluate UAV swarm combat effectiveness but also serve as input samples for machine learning algorithms, supporting the reinforcement learning training of combat decision models. This development opens new avenues for significantly enhancing UAV swarm combat effectiveness.

In recent years, DRL has achieved remarkable breakthroughs in domains such as games, business, and control [3], often surpassing human performance and demonstrating substantial potential for intelligent decision-making applications. Within the military domain, research utilizing DRL is gaining traction: Reference [4] applied heuristic reinforcement learning to air combat intelligent decision-making; Reference [5] employed DRL to study aircraft air-to-ground combat decision-making; Reference [6] implemented cooperative maneuvering decision-making for multiple warheads during penetration using DRL, achieving superior results compared to rule-based methods in simulation; Reference [7] proposed a DRL-based decision-making process framework for multi-aircraft cooperative air combat and validated its feasibility and practicality on a wargaming platform.

This paper first proposes a general modeling methodology for intelligent combat equipment decision-making, integrating combat simulation with DRL. Building upon this, we establish an intelligent decision-making training and modeling framework utilizing the equipment combat simulation system WESS. Subsequently, the paper focuses on the specific problem of combat decision-making modeling for heterogeneous UAV swarm SEAD. We detail the design of an intelligent decision-making model framework, investigate suitable DRL algorithms, and present a case study on intelligent decision-making modeling. The effectiveness of the proposed method and algorithm is validated through experimental comparisons with traditional knowledge engineering-based decision models.

The remainder of this paper is organized as follows. Section II presents the conceptual framework for intelligent equipment combat decision-making modeling based on combat simulation and DRL, as well as the detailed training modeling framework. Section III is dedicated to the decision-making model framework for UAV Swarm SEAD operations, including the operational concept, decision network analysis, and the detailed design of the perception, jamming, and attack decision networks. Section IV describes the training process of the attack decision network based on the Dueling DQN algorithm, covering the algorithm summary, network structure, and reward design. Section V provides a case study to validate the proposed approach, detailing the problem setup, rule experiments, pre-training, iterative training, and intelligent testing results. Finally, Section VI concludes the paper and discusses future work.

II. Intelligent Equipment Combat Decision-Making Modeling Framework Based On Combat Simulation And DRL

A. Conceptual Framework of Intelligent Equipment Combat Decision-Making Model Based on Combat Simulation

Within the framework of combat simulation, the entire intelligent weapon equipment, including its combat decision-making algorithm, must be constructed as a combat simulation model. This enables its incorporation into the combat simulation environment for interactive exploration and learning evolution. The weapon equipment model supporting combat simulation can be divided into two modules based on the operational domain described: the equipment simulation model and the operational behavior model. The former primarily describes behavior within the physical information domain and is responsible for battlefield situation awareness and operational command execution. The latter primarily describes behavior within the cognitive organization domain. It is responsible for analyzing and processing battlefield situation information output by the equipment simulation model, generating action plans, making operational decisions, and passing the resulting commands to the equipment simulation model for execution.

The operational behavior within the cognitive organization domain can be further categorized into two types: pre-war planning behavior and real-time decision-making behavior. The former can be flexibly described using data or scripts and implemented as a scripted operational behavior model within the combat simulation system. The latter requires making ad hoc decisions based on real-time changes in the situation and can be described using various decision-making modeling methods. If knowledge engineering methods are employed for decision modeling, it can be flexibly implemented as a behavior script. If machine learning methods are used for modeling, it is typically implemented as a neural network for inference computation, with DRL used for training and modeling. Current neural network models are generally trained and inferenced using the Python scripting language, enabling their integration into the scripted description framework of combat behavior. During operational simulation, the operational behavior model obtains situation information from the equipment simulation model and generates operational commands based on this information. These commands then drive and control the execution of the equipment simulation model, as depicted in the simulation loop in Figure 1. The training loop shown in the bottom half of the figure indicates that combat simulation generates the training sample data required for DRL-based intelligent decision-making modeling. An updated intelligent decision-making network model is formed through DRL training. The generation of subsequent training sample data is influenced by this updated model rejoining the combat simulation loop, and this process iterates until convergence.

Reinforcement learning algorithms are categorized into two types based on whether the behavior policy and the target policy are identical: On-Policy and Off-Policy. In On-Policy training algorithms, the policy used to generate samples is the same as the policy being optimized. This requires the agent training to be executed synchronously with the combat simulation. Given the computational complexity of combat simulation, On-Policy algorithms are not well-suited for parallel execution of simulations. Conversely, Off-Policy algorithms represent a more suitable training approach. Off-Policy training allows the combat simulation and training processes to run in parallel. Training samples generated from each combat simulation run (termed a round) are written concurrently into the corresponding round's sample database.

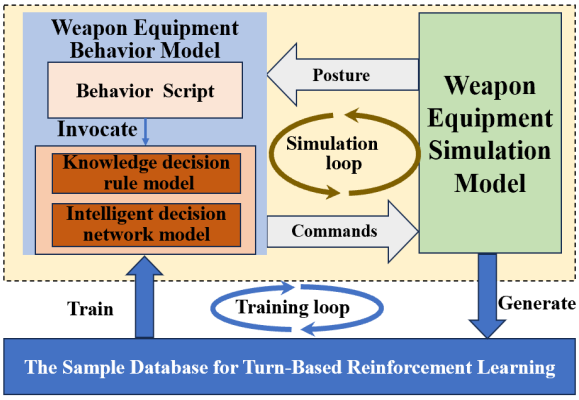

Figure 1. Conceptual framework of intelligent equipment operational decision-making mode

B. Intelligent Equipment Operational Decision-Making Training Modeling Framework Based on DRL.

1) Training modeling process: The process is divided into four stages as shown in Figure 2: rule experiment, pre-training, iterative training, and intelligent comparison test. Suppose there are m intelligent decision-making networks in UAV swarm.

a) The rule experiment: Aims to optimize the decision rules and prepare the pre-training data. By performing Monte Carlo experiments on all the rules of the decision problem in each training scenario space, a large number of rule experiment results data and reinforcement learning round sample data are obtained to evaluate the combat effectiveness of UAV swarm under each combination rule $\pi_R = (R_1, R_2, \dots, R_m)$, and identifies the optimal rule to serve as the benchmark for subsequent intelligent test comparison.

b) The pre-training: Aims to provide an initial network for iterative training. By optimizing the round sample data obtained from the rule experiment, the data set with better combat effect of UAV swarm is obtained. On this basis, each decision network is trained offline to yield $\pi_N^0 = (N_1^0, N_2^0, \dots, N_m^0)$. This stage utilizes the sample data generated by optimal rule experiments to avoid the "cold start" problem in iterative training and improve convergence efficiency.

c) Iterative training: Aims to accumulate experience and improve policy through continuous interaction between

the agent and the training scene. On the basis of the pre-training, the network π_i is iteratively trained in turn, and the rest of the network is fixed in this process. Referring to the idea of policy improvement theorem [8], the policy improvement point is found in the single policy π_i, so that the joint policy π is improved. After all the policies are updated, the single round of policy iteration training is completed until the end conditions are met, such as achieving the desired operational effectiveness index or reaching the maximum number of iteration rounds, etc.The final optimized policy is denoted as $\pi'_N = (N'_1, N'_2, \dots, N'_m)$.

d) The intelligent comparison test: Aims to verify the effectiveness of the single-strategy iteration training. Combat effectiveness evaluations are conducted for the UAV swarm in test scenarios using both the optimized policy π'_N and the baseline policy π_R . The experimental results are then compared to validate the efficacy of the intelligent decision-making approach.

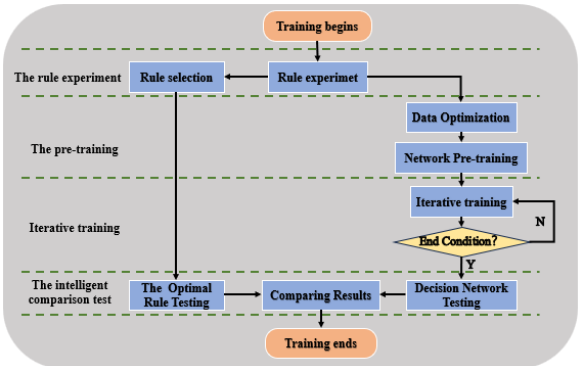

Figure 2. Policy iteration method

2) Training Support Environment : The reinforcement learning training support environment for intelligent decision-making based on combat simulation typically comprises four modules: combat scenario generation tool, combat simulation engine, parallel experiment and training management tool, and reinforcement learning training algorithm. The architecture of this training support environment is depicted in Figure 3.

a) Scenario generation module: Responsible for describing various scenarios that intelligent equipment may encounter in actual combat. It provides the diverse situational data sources required for decision-making model training.

b) Combat simulation engine module [9]*:* Responsible for simulating and executing numerous scenarios, generating both combat effectiveness data and the round sample dataset needed for training.

c) Parallel experiment and training management module: Responsible for managing large-scale parallel simulation experiments. It also orchestrates the synchronous scheduling of the DRL training algorithm and facilitates iterative updates to the decision model during experimentation.

d) Reinforcement learning training module: Responsible for implementing the reinforcement learning algorithm. It accepts scheduling directives from the parallel experiment and training management module and is specifically tasked with generating and updating the decision network model.

III. THE DECISION-MAKING MODEL FRAMEWORK OF UAV SWARM SEAD OPERATION

A. Concept of UAV swarm SEAD operation

In SEAD missions, UAVs must perform reconnaissance, jamming/suppression, and strike tasks autonomously [10]. This enables the swarm to form a complete kill chain and achieve rapid "OODA" cycles. The UAV swarm composition typically includes: a reconnaissance aircraft equipped with radar pods, a jammer with electronic jamming pods, and an attack aircraft armed with multiple anti-radiation missiles.

The typical mission scenario involves: a number of mobile air defense positions (Blue Force) dispersed within a designated area. The Red Force organizes a UAV swarm to conduct SEAD operation against these positions. The attack aircraft form a low-altitude formation. After takeoff from the airfield, they proceed to the periphery of the operational area and enter a holding pattern. The jammer and reconnaissance aircraft form a high-altitude formation. They depart later than the low-altitude formation, flying at ultra-low altitude. At a predefined waypoint, they execute a pop-up maneuver to induce Blue Force air defense radars to activate and reveal their positions. The reconnaissance aircraft then detects and locates these targets, assigning them to the low-altitude attack formation. The primary actions of the attack aircraft (as depicted in Figure 4) are: selecting a launch point upon receiving assigned targets, proceeding to that location, launching missiles, and then entering a holding pattern while awaiting battle damage assessment (BDA) results from the reconnaissance aircraft to determine whether to conduct re-attack or proceed to the next target. The jammer continuously suppresses Blue Force air defense radars and jams incoming missile seekers, creating safer conditions for the low-altitude formation and reconnaissance aircraft. The mission concludes when: the reconnaissance aircraft is destroyed, all attack aircraft missiles are expended, all attack aircraft are destroyed, or all enemy targets are eliminated. Following mission completion, the surviving assets return to base. The objective is to destroy the maximum number of Blue Force air defense radars while sustaining minimum losses.

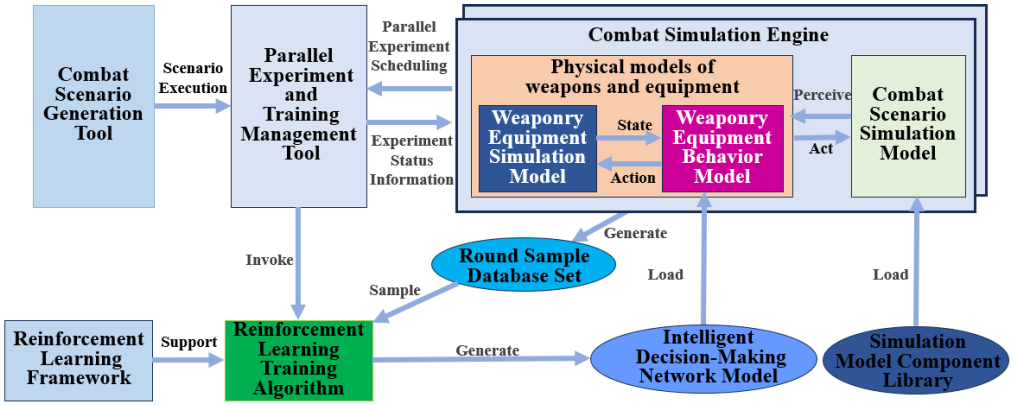

Figure 3. Reinforcement learning training supporting environment for intelligent decision making

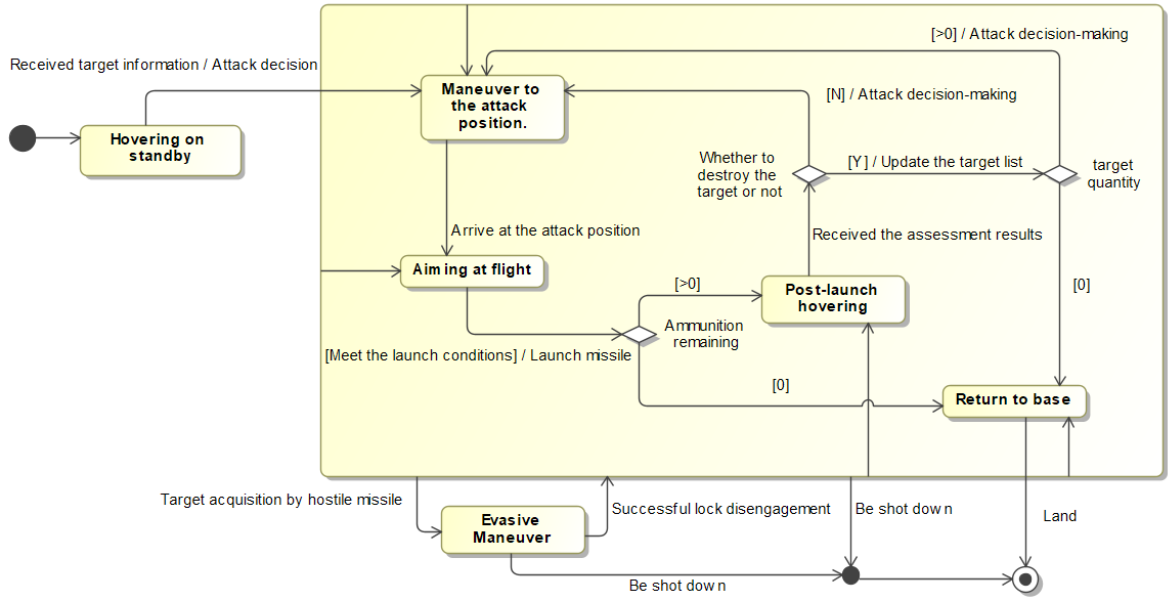

Figure 4. Attack behavior model of attack aircraft

B. *Analysis of UAV swarm SEAD operation decision network*

The SEAD mission encompasses multiple decision-making policies for the UAV swarm, such as formation flying, low-altitude penetration prior to engagement, electronic suppression, detection and perception, fire attack, and target assignment during engagement. Of these, the first three policies (pertaining to the pre-engagement phase) are particularly complex and challenging to describe using rules, and their decision outcomes significantly impact battle results. These decision problems exhibit the characteristics of a Markov decision process, making them suitable for description via neural networks and training using DRL. The remaining policies are directly modeled as rule-based scripts employing knowledge engineering methods. Figure 5 illustrates the composition structure of the entire UAV swarm SEAD operational decision-making model.

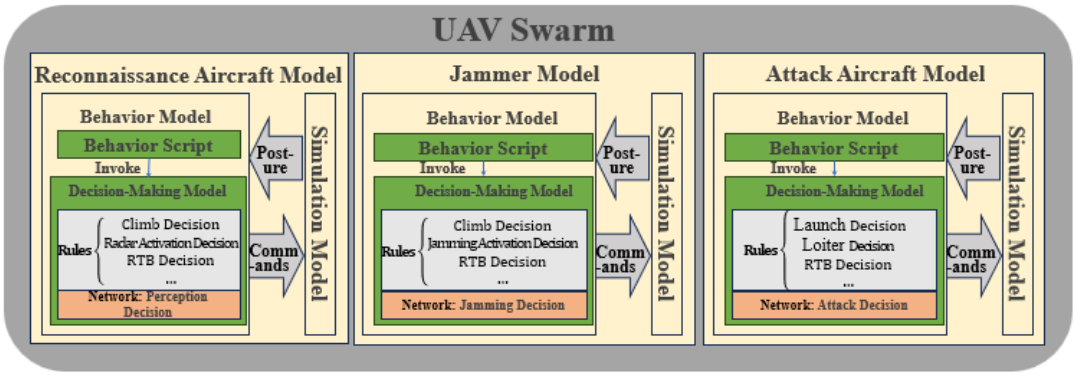

Figure 5. Operational decision-making structure of UAV swarm

C. *UAV swarm SEAD operational decision-making network model framework*

1) *Perception and decision network*

a) Input state space: The perception policy is designed to address the global situational awareness challenge for the reconnaissance aircraft, providing stable targeting information and fire damage assessment to the attack aircraft. It selects the following 4-dimensional inputs:

- Distance and bearing between the reconnaissance aircraft and the target group centroid.
- Distance and bearing between the reconnaissance aircraft and the nearest target.

Given the reconnaissance aircraft's position (x_0, y_0), the target group centroid $(\mathrm{R}_x, \mathrm{R}_y)$ is defined as the weighted average of n targets' coordinates, with weights determined by target importance and threat level.

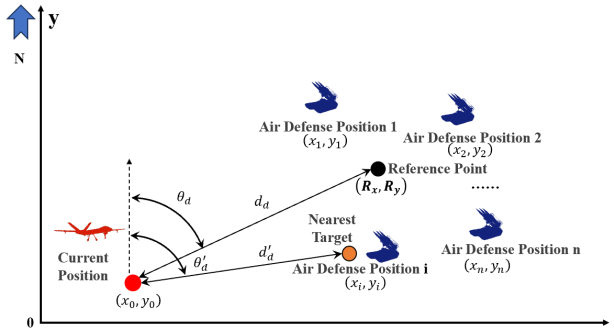

Figure 6. State space analysis of UAV

As illustrated in Figure 6, the distance and azimuth between UAV and the centroid of the target group are described by two parameters, d_d and θ_d, the nearest target is d'_d and θ'_d.Taking the former as an example, it is defined as (1).

$$\begin{cases} d_d = \sqrt{(R_x - x_0)^2 + (R_y - y_0)^2} \\ \theta_d = \tan^{-1}\frac{R_y - y_0}{R_x - x_0} \end{cases} \quad (1)$$

The input state space of the perception decision network can be specifically detailed in Table 1. In practical applications, data preprocessing is performed by taking the ratio of the azimuth value to π and the ratio of the distance value to the radar's maximum detection range (D_{dmax}) for a specific target type, serving normalization purposes.

TABLE I. INPUT STATE SPACE PERCEPTION DECISION NETWORK DESIGN

State variables	*Symbols*	*Data type*	*Preprocessing*
Centroid distance & bearing	d_d、θ_d	double	d/D_{dmax}
Nearest target distance & bearing	d'_d、θ'_d	double	$\theta/180^{\circ}$

b) Action space: In the process of fighting, the different array have different perception, interference effect, so the detection of perception and the output of electronic jamming decision network need to be able to reflect the correlation matrix of information, the concrete can be described as Figure 7.

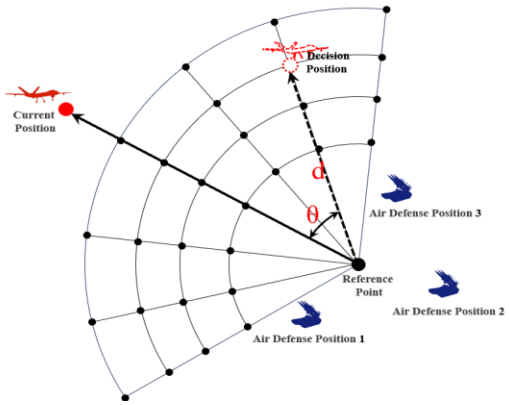

Figure 7. Action diagram of array position selection

Specifically, a polar coordinate system is established with the group centroid as the origin, the vector connecting the centroid to the current sensor array position as the zero-direction, and clockwise orientation as the positive direction; the decision network outputs two-dimensional coordinate information for the array configuration, which undergoes domain-specific processing in operational implementation—including parameter range bounding for direction θ and distance d and value discretization—according to equipment capabilities and mission requirements, for example constraining the operational distance range to 0.5 times D_{dmax} to 1 times D_{dmax} (where D_{dmax} denotes the reconnaissance aircraft's maximum effective detection range) in actual combat scenarios.

TABLE II. THE OUTPUT ACTION SPACE DESIGN OF THE PERCEPTION DECISION NETWORK

Decision-making action variable	*Range of values*	*Notes*
Array position Angle θ	$[-\theta_{max}, \theta_{max}]$	θ_{max} is the maximum Angle delimited
Array position distance d	$[0.5D_{dmax}, D_{dmax}]$	D_{dmax} is the maximum operating distance

c) Call time: The invocation opportunity is:

- When the precise coordinates of the enemy are obtained for the first time.
- When the enemy target is destroyed.
- Attack aircraft was shot down.

2) Interfere with the decision network

a) Interference decision network input state space: The jamming policy aims to solve the problem of enemy suppression and friendly support. The decision network accepts the following state inputs.

- Distance and bearing between the jammer and the centroid of the target cluster.
- Distance and bearing between the jammer and the centroid of the actively engaged target cluster.

The actively engaged target cluster refers to targets currently under attack by strike aircraft. All distance parameters are normalized against D_{jmax} (the jammer's maximum effective jamming range).

TABLE III. INPUT STATE SPACE DESIGN OF JAMMER DECISION NETWORK

State variables	*Symbols*	*Data type*	*Preprocessing*
Real-time attack target group centroid distance, bearing	d_j、θ_j	double	d/D_{jmax} $\theta/180^{\circ}$
Nearest target distance, bearing	d'_j、θ'_j	double	

b) Action space and call timing: During combat operations, both the reconnaissance aircraft and the jammer operate at high altitude. Their situational updates and decision-making actions are synchronized. Consequently, they share an identical action space definition and utilize the same set of call triggers for their respective decision networks.

3) decision network attack

The attack policy aims to solve the attack decision problem of each attack aircraft in the low-altitude formation. The decision network of each attack aircraft is isomorphic, but its execution is asynchronous.

a) Input state space: The attack decision network focuses on the selection of the anti-radiation missile launch position, and selects the following 9-dimensional state information as input (where D_{amax} is the maximum range of the anti-radiation missile and H is the current altitude of the attack aircraft):

- The distance, azimuth, and altitude difference between the current position of the attack aircraft and the target.
- The distance, azimuth, and elevation difference between the current position of the attack aircraft and the maximum threat target (the air defense position closest to the current attack target); the launch position should avoid this threat as much as possible.
- The distance and azimuth between the jammer and the attack target; the suppression effect of the jammer varies with its position.
- The current attack round count for the target and a flag indicating whether the first attack on this target was successful, reflecting the target's defensive capability strength or weakness.

All input data undergoes normalization.

TABLE IV. ATTACK DECISION NETWORK INPUT STATE SPACE DESIGN

State variables	*Symbols*	*Data type*	*Preprocessing*
Target distance, azimuth, elevation difference	$\boldsymbol{d_a, \theta_a, h_a}$	double	d/D_{amax} $\theta/180$ h/H
Maximum threat distance, bearing, altitude difference	d'_a, θ'_a, h'_a	double	
Jammer range, bearing	d_{aj}, θ_{aj}	double	
Current attack round	n	int	——

b) Output action space: The outcome of the attack decision network is the relative launch position of the attack aircraft with respect to the current target, defined by the distance and bearing between the launch position and the target. This concept mirrors the decision outputs of the reconnaissance aircraft. Based on the operational range of the anti-radiation missile and the capabilities of the air defense systems, the valid ranges for these parameters are defined as $[D_{amin}, D_{amax}]$ or distance and $[-\theta_{amax}, \theta_{amax}]$ for bearing. These ranges are discretized into $N_d + 1$ points (i.e., endpoints of N_d segments) for distance and $N_{\boldsymbol{\theta}} + 1$ points (i.e., endpoints of $N_{\boldsymbol{\theta}}$ segments) for bearing, forming a total of $N_{d\theta}$ discrete actions. The decision network evaluates the value of each discrete action and selects the one with the highest value as the optimal decision. This selected action is then converted into precise coordinates for the launch position. The attack aircraft maneuvers to this position to execute the strike.

TABLE V. ATTACK DECISION NETWORK OUTPUT SPACE DESIGN

Decision action variables	*Range of values*	*Action parsing*
Launch Position N	$\{0,1,\dots,N_{d\theta}\}$	$d = \frac{(D_{amax} - D_{amin})}{N_d} \times \left\lfloor \frac{N}{(N_\theta + 1)} \right\rfloor + D_{amin}$ $\theta = \frac{2\theta_{amax}}{N_\theta} \times \left(N \bmod (N_\theta + 1)\right) - \theta_{amax}$

c) Call timing: The invocation of the attack decision network is centered on the attack target and requires the anti-radiation missile's passive seeker to lock onto a stable enemy radar beam to ensure attack accuracy. Therefore, subject to the preconditions of nonzero remaining ammunition and the reconnaissance aircraft being operational, the attack decision network is triggered under the following conditions:

- When the coordinates of the attack target are obtained for the first time.
- Upon attack failure.
- When successfully switching the attack target after a previous attack.
- Training the attack decision network based on Dueling DQN.

IV. TRAINING THE ATTACK DECISION NETWORK BASED ON DUELING DQN

To illustrate the training of the attack decision network as an example of intelligent decision-making, the remaining tactics employ the optimal rules.

A. Summary of Algorithm

Within the algorithm framework selection, the decision networks for each attack aircraft in the low-altitude formation are completely homogeneous; that is, they share identical state spaces, action spaces, and optimization objectives. This scenario can be simplified as a single-agent decision problem.

For the specific algorithm, addressing high-dimensional input, large action spaces, and the need to finely distinguish state and action values, the Dueling Deep Q-Network (Dueling DQN [11]) method demonstrates strong performance. As an improved algorithm over DQN [12], its core innovation is the decomposition of the traditional Q-value into two components: the state value $V(s)$ and the action advantage $A(s,a)$. A dual-branch neural network structure is employed to learn these two parts separately. The final action value $Q(s,a)$ is then calculated using the combination formula (2) (where $|\mathcal{A}|$ represents the size of the action space):

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a' \in \mathcal{A}} A(s,a')\right) \quad (2)$$

This design enables the model to more effectively capture the relationship between state and action. It is particularly suitable for environments where the state value remains relatively stable while action advantages exhibit significant variation, thereby improving the algorithm's learning efficiency and stability.

B. Network Structure

The Dueling DQN algorithm is value-based. Its neural network architecture comprises two Q-networks with identical structures: a training network updated in real-time and a target network. The target network parameters are periodically synchronized with the training network parameters every fixed number of steps to enhance training stability. As shown in Figure 8, the Q-network utilizes a fully connected neural network that takes the 9-dimensional state vector as input and outputs Q-values for 35 discrete actions. Actions are selected according to a greedy policy, choosing the action with the maximum Q-value. This structure enables efficient feature sharing and effectively captures the dynamic advantages of different actions, making it particularly impactful in large-scale discrete action spaces and well-suited for the task.

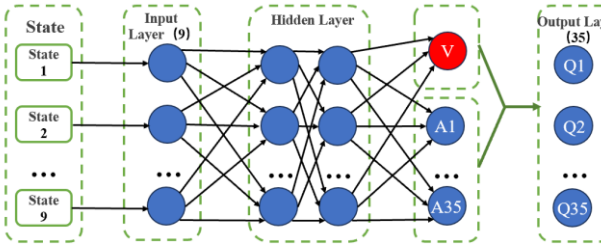

Figure 8. Dueling DQN decision network structure

C. Reward Design

The reward function plays a crucial role in guiding the iterative improvement of the decision-making network. Aligned with the objective of maximizing the exchange ratio and prioritizing the protection of the reconnaissance aircraft, the reward function evaluates operational actions based on the following dimensions: 1) reconnaissance aircraft survival status; 2) attack aircraft survival status. 3) successful missile launch; 4) successful missile hit on target. 16 distinct operational states are defined, encompassing nine feasible combinations of these dimensions. To differentiate between initial and supplementary attacks, a unique reward value is assigned to each state.

TABLE VI. REWARD DESIGN TABLE

No	State	Reward Value		No	State	Reward Value		No	State	Reward Value	
		Initial	*Follow-up*			*Initial*	*Follow-up*			*Initial*	*Follow-up*
1	[0,0,1,0]	-60	-65	4	[0,1,1,1]	-2	-4	7	[1,1,0,0]	0	0
2	[0,1,0,0]	-80	-90	5	[1,0,0,0]	-70	-75	8	[1,1,1,0]	20	10
3	[0,1,1,0]	-20	-40	6	[1,0,1,0]	-5	-10	9	[1,1,1,1]	90	50

TABLE VII. FORCES

Force	Units	Primary Mission Payload	Force	Units	Primary Mission Payload
Red Force	Reconnaissance Aircraft ×1	Radar Pod ×1	Blue Force	Early Warning Site ×1	Early Warning Radar ×1
	Jammer Aircraft ×1	Jamming Pod ×1		Air Defense Positions ×3	Fire Control Radar ×1
	Attack Aircraft ×2	Anti-Radiation Missiles ×2			Surface-to-Air Missiles ×24

The states [0,0,0,1], [0,1,0,1], [1,0,0,1], and [1,1,0,1] represent "missile launched but not hit" – these are impossible because if an attack aircraft is shot down after launch, the missile outcome becomes unknowable within the simulation. The states [1,0,1,1] and [0,0,1,1] are also impossible because the mission turn terminates immediately upon destruction of either the reconnaissance aircraft or an attack aircraft, precluding subsequent missile impact assessment. Additionally, the state [0,0,0,0] (indicating no launch and no hits) is impossible as it contradicts the context of evaluating attack actions.

V. SEAD UAV SWARM DECISION TRAINING MODEL

In the case study design and implementation, the training and verification environment was constructed using the WESS system developed by the research team [13].

A. Case Problem

The baseline forces for both Red and Blue sides are configured as detailed in TABLE VII. The lethality parameters are defined as follows: one anti-radiation missile is assumed to paralyze an air defense position, and one surface-to-air missile is assumed to shoot down one UAV.

For the Blue Force, early warning radars and air defense positions operate as an integrated system. In the absence of enemy threats, air defense radars remain silent to conceal their positions, while early warning radars—with longer detection ranges and wider scanning fields—perform aerial surveillance. Upon detecting incoming strikes, the early warning system relays target information to air defense positions in real time. When activated, air defense positions power on fire control radars for aerial search. After target lock is achieved and launch readiness confirmed, they intercept aerial targets (aircraft or missiles) using either autonomous or third-party guidance. If a target is destroyed and additional threats remain, engagement continues; if a missile misses, immediate re-engagement is initiated. Combat concludes when the position is destroyed or all airborne threats are neutralized.

Scenario generation involves six key variables defining the initial disposition of three enemy air defense positions. Each dimension (representing longitudinal or latitudinal coordinate offsets from actual deployment locations) has two discrete values, resulting in 64 distinct scenario configurations.

TABLE VIII. SCENARIO GENERATION COMBAT SCENARIO

Intelligence Position	Longitude Offset (minutes)	Latitude Offset (minutes)
Air Defense Company 1 (lon1, lat1)	$\{-1.58, 1.59\}$	$\{-2.04, 2.04\}$
Air Defense Company 2 (lon2, lat2)	$\{-1.87, 1.91\}$	$\{-1.94, 2.00\}$
Air Defense Company 3 (lon3, lat3)	$\{-1.95, 1.95\}$	$\{-2.27, 2.22\}$

B. Rule Experiments

The launch distance range [20,40] km was discretized into 5 values and the azimuth range $[-30,30]$ degrees into 7 values, generating 35 candidate launch positions. Each position was evaluated across all 64 scenarios with 10 Monte Carlo repetitions per configuration. Combat effectiveness was quantified using the average exchange ratio per launch position over all experimental runs. This resulted in 35×64=2,240 scenario-position combinations and 22,400 total simulation runs. Key findings, visualized in the heat map of Figure 9, are summarized below:

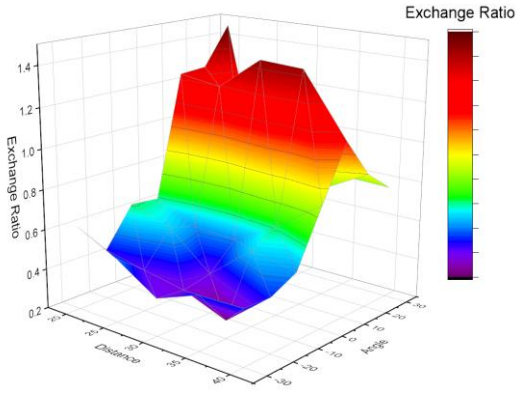

Figure 9. Specific heat map of regular experiment exchange

A The results demonstrate that selecting the launch position 20 km from the target at a 30° offset azimuth delivers optimal performance, achieving an average exchange ratio of 1.465.

C. Pre-training

1) Hyperparameter configuration : The value network was implemented using PyTorch. Core algorithm parameters include:

TABLE IX. DECISION NETWORK ATTACK DUELING DQN TRAINING PART PARAMETER CONFIGURATION

Hyperparameter	Value
Decision Network	2 hidden layers with 128 and 64 units
Discount Factor	0.99
Discount Factor	ReLU
Learning Rate	0.0001
Experience Replay Buffer Size	100000
Batch Size	256
Delayed Update Steps	200

2) Pre-train : Utilizing data samples where the reward value was non-negative, the neural network was trained for 100 rounds. Each round consisted of 200 training steps. Upon network convergence, the resulting model served as the initial decision model.

D. Iterative Training of Reinforcement Learning

he pre-trained attack decision-making model was loaded. For each training scenario, 10 simulation runs were executed, and the resulting experience data were stored in the database. The reinforcement learning algorithm then extracted batches of experience data from this database to train the decision network. After each training update, the updated decision network was loaded back into the simulation environment. This process—completing all scenario experiments—constituted one training round. The exploration rate was decayed by 0.01 per round. Training continued iteratively until the reward signal stabilized.

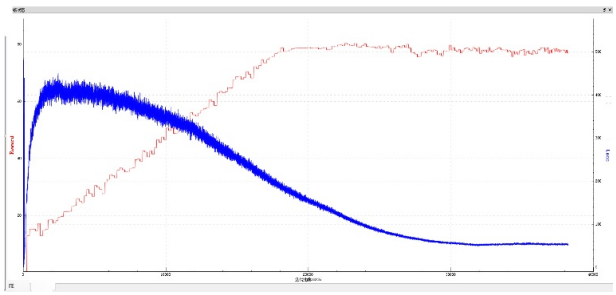

Figure 10. Schematic figure 10 reward (red curve) and loss (blue curve)

As shown in Figure 10, the average reward per round (calculated over 640 simulation runs) and the loss (mean squared error (MSE) between predicted and target Q-values) converged over the training process. After 38,235 training iterations spanning 200 rounds, the results stabilized: the average reward plateaued around 79. In the training scenarios, the attack aircraft demonstrated effective decision-making, achieving successful target hits both during initial engagements and follow-up attacks, while significantly improving the survival rate of the reconnaissance aircraft.

E. Intelligent Testing

The pre-trained attack decision-making model was loaded. For each training scenario, 10 simulation runs were executed, and the resulting experience data were stored in the database. The reinforcement learning algorithm then extracted batches of experience data from this database to train the decision network. After each training update, the updated decision network was reloaded into the simulation environment. This process—completing experiments across all scenarios—constituted one training round. The exploration rate was decayed by 0.01 per round. Training continued iteratively until the reward signal stabilized.

As shown in Figure 11, a total of five distinct test scenarios were constructed. The target locations of the three air-defense batteries remained consistent across scenarios. The actual deployment positions within each scenario corresponded to the vertices of the depicted rectangles. Brown markers represent training scenario positions, while other colors denote test scenario positions. To ensure simulation fidelity, each air-defense battery possessed four distinct disposition patterns. The combination of these patterns across the three batteries generated 64 unique Blue Force deployment configurations.

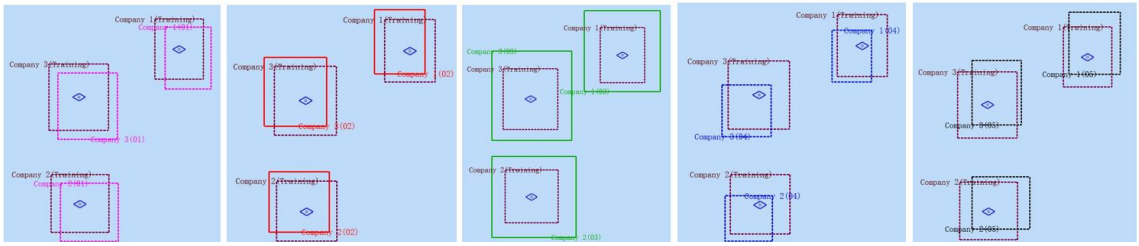

Figure 11. Intelligence test scenario 1-5

TABLE X. COMBAT SCENARIO TESTING SCENARIO GENERATION DESIGN

Scenario	Company 1		Company 2		Company 3	
	Longitude Offset	*Latitude Offset*	*Longitude Offset*	*Latitude Offset*	*Longitude Offset*	*Latitude Offset*
1	{−0.90,2.10}	{−2.70,1.50}	{−1.26,2.52}	{−2.54,1.40}	{−1.38,2.52}	{−2.87,1.63}
2	{−2.22,0.96}	{−1.50,2.70}	{−2.34,1.44}	{−1.34,2.60}	{−2.58,1.32}	{−1.67,2.83}
3	{−2.70,2.70}	{−2.70,3.30}	{−2.82,3.18}	{−3.00,3.00}	{−2.76,2.94}	{−3.06,3.54}
4	{−1.92,0.60}	{−2.40,1.02}	{−2.22,2.52}	{−2.40,0.60}	{−2.34,0.78}	{−2.76,0.66}
5	{−1.20,2.16}	{−1.20,3.06}	{−1.08,2.70}	{−1.20,2.34}	{−1.02,2.22}	{−1.38,3.00}

TABLE XI. STATISTICS OF STRIKE EFFECTS

Engagement Policy	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Optimal Rule-Based Policy	1.286765	1.102233	0.853608	1.276376	0.934328
Intelligent Decision Policy	1.503086	1.383281	1.039818	1.765705	1.19802
Performance Improvement	16.81%	25.50%	21.81%	38.34%	28.22%

For each of the 64 deployment configurations, 10 simulation runs were conducted. The resulting damage data for both Red and Blue forces across all 640 experiments were recorded and aggregated to calculate the exchange ratio.

As can be seen from the experimental results, in the new scenario, the exchange of the intelligent decision rules are better than decisions, verify the advantages of the intelligent decision.

VI. CONCLUSION

This paper addresses the intelligent decision-making problem for UAV Swarm SEAD missions. A decision modeling approach combining DRL with combat simulation is proposed, a corresponding modeling framework is constructed, and attack decision modeling based on the Dueling DQN algorithm is implemented. Leveraging the structured WESS system as a reinforcement learning training environment and designing a case study, experimental results verify that the DRL-based intelligent decision-making approach yields superior decision quality compared to traditional knowledge engineering-based methods.

For UAV Swarm SEAD tasks, this paper designs a comprehensive simulation scenario framework, successfully integrates the intelligent decision model, and demonstrates the feasibility of the DRL method. This work provides a training environment for subsequent intelligent decision-making research concerning reconnaissance and jammer aircraft within the swarm. Furthermore, it contributes to enhancing the overall intelligence level of UAV Swarms in SEAD missions and offers valuable insights for UAV Swarm decision modeling in other operational scenarios.

REFERENCES

[1] J. Tang, X. Li, and J. Dai, "Analysis on the application of U.S. UAV Air Defense Suppression Operations," *Aerodynamic Missile Journal*, 2020, no. 5, pp. 44-48.

[2] D. D. Diehl, *How to optimize joint theater ballistic missile defense*, M.S. thesis, Naval Postgraduate School, Monterey, CA, USA, 2018.

[3] S. Yang, Z. Shan, Y. Ding, and G. Li, "Survey of research on deep reinforcement learning," *Computer Engineering*, 2021, vol. 47, no. 12, pp. 19-29.

[4] J. L. Zuo, R. N. Yang, Y. Zhang, Z. Li, and M. Wu, "Intelligent decision-making in air combat maneuvering based on heuristic reinforcement learning," *Acta Aeronautica et Astronautica Sinica*, 2017, vol. 38, no. 10, p. 321168.

[5] L. Zhang, J. Xu, et al., "Air Dominance Through Machine Learning: A Preliminary Exploration of Artificial Intelligence - Assisted Mission Planning," RAND Corporation, 2020. [Online]. Available: https://doi.org/10.7249/RR4311 [retrieved: Sept, 2025].

[6] Y. Wang, Y. Lei, S. Lei, et al., "Research on multi-warhead Cooperative penetration Decision Modeling Based on Deep Reinforcement Learning," in *Proc. The Third Conference on Systems Engineering-Complex Systems and Systems Engineering Management*, 2021-04-16.

[7] W. Shi, Y. Feng, G. Cheng, H. Huang, J. Huang, Z. Liu, and W. He, "Research on multi-aircraft cooperative air combat method based on deep reinforcement learning," *Acta Automatica Sinica*, 2021, vol. 47, no. 7, pp. 1610-1623.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018, ch. 4.2.

[9] F. Lu, X. Hu, B. Zhao, X. Jiang, D. Liu, J. Lai, and Z. Wang, "Review of the Research Progress in Combat Simulation Software," *Applied Sciences*, 2023, vol. 13, no. 9, p. 557.

[10] X. Wang, J. Cheng, Q. Guo, S. He, F. Guo, and Y. Chen, "Research on Intelligent Cooperative Combat System in Air Defense suppression Mission," *Unmanned System Technology*, 2020, vol. 3, no. 4, pp. 10-21.

[11] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," *arXiv preprint arXiv:1511.06581*, 2015. [Revised: Apr. 2016]. Available: https://arxiv.org/abs/1511.06581 [retrieved: Sept, 2025].

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *NIPS Deep Learning Workshop*, 2013. [Online]. Available: https://arxiv.org/abs/1312.5602 [retrieved: Sept, 2025].

[13] Y. Lei, J. Yao, N. Zhu, Y. Zhu, and W. Wang, "Weapon Equipment Combat Effectiveness Simulation System WESS," *Journal of System Simulation*, 2017, vol. 29, no. 6, pp. 1244-1252.