Artificial Intelligence in Combat Decision-making: Weapon Target Assignment via Reinforcement Learning and Graph Neural Networks

Seung Heon Oh[1], Geon Woong Byeon[1], Young In Cho[1], Seungmin Kwon[1], and Jong Hun Woo[1]

[1]Affiliation not available

March 04, 2025

Abstract

Selecting a threat to attack is one of the most important decisions on the battlefield. The decision problem is represented as a Weapon-Target Assignment problem (WTA) problem. In the previous studies, dynamic programming, linear programming, metaheuristics, and heuristic methods have been applied to solve this problem. However, previous studies have been limited by oversimplified-model, computational burden, lack of adaptability to disruptive events, and recalculation when the problem size changes. To overcome these limitations, this study aims to solve WTA by using reinforcement learning and graph neural networks. The proposed method has high practicality by reflecting the real-world decision-making framework, OODA-loop (Observe-Orient-Decide). Experiments are conducted in various environments, and the effectiveness of the proposed method is demonstrated by comparing it with existing heuristic and meta-heuristic methodologies. The proposed method introduces a groundbreaking methodology for intelligent decision-making in tactical command and control traditionally considered exclusive to human-expert.

Artificial Intelligence in Combat Decision-making: Weapon Target Assignment via Reinforcement Learning and Graph Neural Networks

Seung Heon Oh, Geon Woong Byeon, Young In Cho, Seungmin Kwon and Jong Hun Woo

***Abstract*—Selecting a target to attack is one of the most critical decisions on the battlefield. The decision problem is represented as a dynamic weapon-target assignment (DWTA) problem. While deep reinforcement learning (DRL) is the state-of-the-art approach for DWTA, previous studies have limitations in three key aspects: representing topological relationships on the battlefield, scalability to increased problem sizes, and the practicality of the objective function. To overcome these limitations, this study aims to solve the DWTA problem by leveraging DRL and graph neural networks (GNN), with a novel partially observable Markov decision process (POMDP) design including graph-based action representation, observation feature, and reward structuring. Experiments are conducted across multiple military domains, including naval and ground combat, comparing the proposed approach with existing heuristic and meta-heuristic methodologies. The effectiveness of the GNN and decision-making pattern is extensively analyzed through comprehensive experimental validation.**

***Index Terms*—Weapon Target Assignment Problem, Reinforcement Learning, Graph Neural Network**

I. Introduction

Combat commanders must make decisions under extreme uncertainty, which stems from incomplete enemy information and unpredictable events. The OODA (Observe-Orient-Decide-Act) loop emphasizes that combat commanders must rapidly adapt their decision-making to evolving battlefield conditions through cyclic information processing and action under uncertainty. Weapon Target Assignment (WTA), a key element in combat decision-making, is an NP-hard combinatorial optimization problem [1] that must be aligned with the OODA framework for practical implementation [2].

Following the OODA loop concept, WTA must be treated as a dynamic optimization problem (i.e. Dynamic WTA, DWTA) that reflects the time-evolving environment [3] (see supplementary material A). Traditional approaches to solving DWTA are open-loop methods, which optimize solutions based solely on initial conditions. [4] applied a greedy algorithm, and [5] utilized stochastic programming to solve WTA problems, while [6] combined greedy heuristics with nonlinear network flow. However, these open-loop approaches have limitations in adapting to rapidly evolving combat situations. They require computationally expensive replanning to react to unpredictable or stochastic events such as new threat insertion, decoying event, or target hit. Their computational inefficiency contradicts the OODA loop's rapid decision-making principle.

To address this issue, the closed-loop approach performs real-time decision making, which includes methods such as exact, two-stage, and heuristic approaches. Exact methods like dynamic programming [7] and mixed-integer linear programming [8] adopt state-based sequential decision-making. Despite their optimality guarantee, they face curse of dimensionality and computational burden. Meta-heuristics offer efficient alternatives, with [3] combining constructive heuristics and tabu search, and [9] applying genetic algorithm (GA). Adopting anytime frameworks, meta-heuristic methods gradually improve solutions until reaching time user-defined limits, allowing real-time implementation. Two-stage approaches decompose WTA into sequencing and assignment problems to enhance the computational efficiency. [10] adopts the Hungarian algorithm for assignment and particle swarm optimization for sequencing. Both meta-heuristic and two-stage methods remain sensitive to computation time and problem scale. Heuristic approaches [11], [12], [13] provide quick adaptation with minimal computation, despite suboptimal solutions. Notably, recent studies [13], [14] emphasize the integration of high-fidelity wargame simulations to enhance real-world applicability beyond lab-scale combinatorial optimization research.

Deep Reinforcement Learning (DRL) approaches, including value-based DRL [15], deep Q-network (DQN) [16], and actor-critic methods [17], have emerged as state-of-the-art (SOTA) solution for DWTA, offering significant advantages in wargame simulation integration, real-time adaptability [18] and long-term reward optimization under uncertainty [19], [20].

However, they do not fully account for the topological relationships in real-world battlefields, such as attacking [21], [22] and engagement availability [23], [5]. Previous studies have represented these topological relationships through vectors [15], [16], [24], images [17], or sequences

Seung Heon Oh, Gun Woong Byeon, Young In Cho, Jong Hun Woo are with the Department of Naval Architecture and Ocean Engineering, Seoul National University, Seoul, 08826, Republic of Korea (e-mail: j.woo@snu.ac.kr)

Jong Hun Woo is with Research Institute of Marine Systems Engineering, Seoul National University, Seoul 08826, Korea.

Seungmin Kwon is with Naval Ship R&D Team of Hanwha Ocean Co., Ltd., 3370, Geoje-daero, Geoje-si, Gyeongsangnam-do, 53302, Republic of Korea (e-mail: smkwon@hanwha.com)

[25], which might fail to capture the valuable information essential for quality decision-making that is latent in topological relationships (representation issue) [26]. Furthermore, these studies primarily focus on maximizing the cumulative value of destroyed targets, which is difficult to accurately assess in real-world combat situations, rather than maximizing platform survivability, which is both measurable and aligns with the emphasis on damage minimization in modern warfare [3] (practicality). To address these limitations, we make the following contributions:

- To overcome representation issue, we propose a novel architecture that integrates DRL with Graph Neural Network (GNN) to enhance the representation of topological relationships inherent in real-world battlefields. To the best of our knowledge, this is the first approach that combines DRL and GNN in the WTA domain, and our empirical results show that this integration leads to substantial performance gains over SOTA DRL approaches.
- To overcome scalability issue, we develop a graph-based action representation framework that enables scalability across different problem sizes. This framework integrated into GNN maintains consistent performance without retraining when the number of units and threats vary.
- To enhance practicality, we propose novel observation features and reward design that contribute to maximizing platform survival probability, enabling comprehensive battlefield awareness and efficient policy optimization, respectively.
- The proposed method is integrated and validated with high-fidelity wargame simulations across multiple military domains (naval and ground combat). This extensive validation demonstrates that our approach goes beyond lab-scale combinatorial optimization, establishing its viability for real-world military applications.

Nomenclature

Index

$e_{i,q}$	q-th sub-process of W_i
i	Index for interceptor
j	Index of threat
K_j	Threat with index j
q	Sequence index of interceptor sub-process
t, h, τ	Decision timestep; $t, h, \tau \in \{1, 2, ...T-1\}$
W_i	Interceptor i

Binary variable

$w_{ij}(t,h)$	Simultaneous guidance seizing indicator. The physical meaning of $w_{ij}(t,h)$ indicates whether a missile W_i launched at timestep h targeting K_j is seizing the simultaneous guidance resource at timestep t: if $t \leq h + \sum_{q=1}^{N_{\mathbf{E}_i}} o_{ijq}(h)$, then $w_{ij}(t,h) = 1$; otherwise, $w_{ij}(t,h) = 0$.
$x_{ij}(t)$	Decision variable whether to launch the missile of W_i towards K_j at timestep t: if selected to launch, then $x_{ij}(t) = 1$; otherwise, $x_{ij}(t) = 0$.
$y_{ij}(t,\tau)$	Arrival-on-target indicator. The phyisical meaning of $y_{ij}(t,\tau)$ is whether a missile of W_i launched at timestep t arrives on K_j at timestep τ: if $\tau = t + a_{ij}(t)$, then $y_{ij}(t,\tau) = 1$; otherwise, $y_{ij}(t,\tau) = 0$.
$z_{ij}^{(1)}(t,\tau), z_{ij}^{(2)}(t,\tau)$	Non-arrival indicators. These variables indicate when a missile from W_i launched at timestep t does not arrive at K_j at timestep τ. When $y_{ij}(t,\tau) = 1$, both $z_{ij}^{(1)}(t,\tau)$ and $z_{ij}^{(2)}(t,\tau)$ are 0; otherwise, they are 0.

Integer variable

$a_{ij}(t)$	Flight duration variable : if W_i launches a missile at timestep t targeting K_j, then $a_{ij}(t) = a_{ijt}$; otherwise, $a_{ij}(t)$ is sufficiently large number
$o_{ijq}(t)$	Simultaneous guidance seizing duration variable: if W_i launches a missile at timestep t targeting K_j, then $o_{ijq}(t) = o_{ijqt}$; otherwise, $o_{ijq}(t) = 0$

Parameters

δ	Sufficiently small positive number
γ_j	Probability of K_j hitting the own asset upon its arrival
a_{ijt}	Flight time for missile of W_i launched at timestep t until arriving at K_j
$b_{ij}^{(1)}$	Earliest time when W_i can intercept K_j
$b_j^{(2)}$	Arrival time of K_j to the own asset
M	Sufficiently large number
o_{ijqt}	Processing time of $e_{i,q}$: time required to complete the qth sub-process when missile W_i is launched against K_j at timestep t
p_{ij}	Probability that a missile launched from W_i hits K_j upon its arrival
T	Terminal time

II. Problem Definition

In this study, we focus on the one-on-many DWTA settings (one asset versus multiple threats), aiming to maximize the asset's survivability. In the problem, the number of interceptor's type that own asset has is $N_{\mathbf{W}}$. Interceptor W_i belongs to the set of interception systems $\mathbf{W} = \{W_1, W_2, \ldots, W_{N_{\mathbf{W}}}\}$. The number of missiles of W_i is P_i (i.e., up to P_i missiles can be launched). Threat K_j belongs to the threat set $\mathbf{K} = \{K_1, K_2, \ldots, K_{N_{\mathbf{K}}}\}$. There are two categories of threats: launcher-types and missile-types, where missile-types are launched by launcher-types. W_i can only intercept threats K_j that belong to the set of engagement available set $\mathbf{A}_i$. The simultaneous guidance resource type set $\mathbf{M}$ is the set whose elements are the set of the simultaneous guidance resource[1] set. The simultaneous guidance resource set $\mathbf{M}_k$ is an element of $\mathbf{M}$: $\mathbf{M}_k \in \mathbf{M}$. W_i belong to one of the simultaneous guidance resource set $\mathbf{M}_k$ within the set $\mathbf{M}$: $\exists! k, W_i \in \mathbf{M}_k, \mathbf{M}_k \in \mathbf{M}, \forall W_i \in \mathbf{W}$. The simultaneous guidance capacity of $\mathbf{M}_k$ is C_k, which means that C_k missiles launched by interceptors belonging to $\mathbf{M}_k$ can be guided simultaneously. Ordered tuple $\mathbf{E}_i$ is a series of $N_{\mathbf{E}_i}$ sub-processes which the missile launched

[1]It is related to the management capacity of the operator or the specification of the combat system.

by W_i must complete until releasing its simultaneous guidance resources: $\mathbf{E}_i : (e_{i,1}, e_{i,2}, \ldots, e_{i,N_{\mathbf{E}_i}})$ (the simultaneous guidance resource is seized immediately after the launch command is given, starting $e_{i,1}$ and ending $e_{i,N_{\mathbf{E}_i}}$ and the simultaneous guidance resource is released when $e_{i,N_{\mathbf{E}_i}}$ ends.). The objective function aims to maximize the survivability of our own assets by minimizing the probability of being hit by threats.

$$\max \quad J(\mathbf{X}) = \prod_{K_j \in \mathbf{K}} (1 - \gamma_j \prod_{t=1}^{T-1} \prod_{\tau'=t}^{T} \prod_{W_i \in \mathbf{W}} (1 - p_{ij})^{y_{ij}(t,\tau')}) \tag{1}$$

$$\text{s.t.} \quad \sum_{\tau'=1}^{t} \sum_{K_j \in \mathbf{K}} x_{ij}(\tau') \leq P_i; \quad \forall i, t, W_i \in \mathbf{W} \tag{2}$$

$$\sum_{W_i \in \mathbf{W}} \sum_{K_j \in \mathbf{K}} x_{ij}(t) \leq 1; \quad \forall t \tag{3}$$

$$x_{ij}(t) = 0; \quad \forall i, j, W_i \in \mathbf{W}, K_j \notin \mathbf{A}_i \text{ or } t \in \{t : t > b_{ij}^{(1)} \text{ or } t \leq b_j^{(2)}\} \tag{4}$$

$$a_{ij}(t) - M(1 - x_{ij}(t)) - a_{ijt} = 0; \quad \forall i, j, t, W_i \in \mathbf{W}, K_j \in \mathbf{K} \tag{5}$$

$$\tau \leq (t + a_{ij}(t) - \delta) z_{ij}^{(1)}(t,\tau) + (t + a_{ij}(t)) y_{ij}(t,\tau) + M z_{ij}^{(2)}(t,\tau); \quad \forall i, j, t, \tau, W_i \in \mathbf{W}, K_j \in \mathbf{K} \tag{6}$$

$$\tau \geq t z_{ij}^{(1)}(t,\tau) + (t + a_{ij}(t)) y_{ij}(t,\tau) + (t + a_{ij}(t) + \delta) z_{ij}^{(2)}(t,\tau); \quad \forall i, j, t, \tau, W_i \in \mathbf{W}, K_j \in \mathbf{K} \tag{7}$$

$$y_{ij}(t,\tau) + z_{ij}^{(1)}(t,\tau) + z_{ij}^{(2)}(t,\tau) = 1; \quad \forall i, j, t, \tau, W_i \in \mathbf{W}, K_j \in \mathbf{K} \tag{8}$$

$$o_{ijq}(t) - o_{ijqt} x_{ij}(t) = 0; \quad \forall i, j, q, t, W_i \in \mathbf{W}, e_{i,q} \in \mathbf{E}_i, K_j \in \mathbf{K} \tag{9}$$

$$h + \sum_{q=1}^{N_{\mathbf{E}_i}} o_{ijq}(h) - t \leq M w_{ij}(t,h) - \delta \quad \forall i, j, W_i \in \mathbf{W}, K_j \in \mathbf{K} \quad \forall t, \forall h \leq t \tag{10}$$

$$-(h + \sum_{q=1}^{N_{\mathbf{E}_i}} o_{ijq}(h) - t) \leq M(1 - w_{ij}(t,h)) \quad \forall i, j, W_i \in \mathbf{W}, K_j \in \mathbf{K} \quad \forall t, \forall h \leq t \tag{11}$$

$$\sum_{W_i \in \mathbf{M}_k} \sum_{K_j \in \mathbf{A}_i} \sum_{h=1}^{t} w_{ij}(t,h) \leq C_k; \quad \forall k, t, \mathbf{M}_k \in \mathbf{M} \tag{12}$$

$$x_{ij}(t), y_{ij}(t,\tau), w_{ij}(t,h), z_{ij}^{(1)}(t,\tau), z_{ij}^{(2)}(t,\tau) \in \{0,1\}; \quad \forall i, j, t, \tau, h, W_i \in \mathbf{W}, K_j \in \mathbf{K} \tag{13}$$

The problem is formulated as a non-linear integer programming, which follows as (1)-(13). $x_{ij}(t)$ is a binary decision variable indicating whether to launch interceptor i (W_i) towards threat j (K_j) at timestep t. The indices i and j represent the interceptor types and threats respectively, while t denotes the decision timestep within the planning horizon $\{1, 2, ..., T-1\}$. When $x_{ij}(t) = 1$, it indicates a launch decision, and $x_{ij}(t) = 0$ indicates no launch. $\mathbf{x}(t) = [x_{11}(t), ..., x_{N_{\mathbf{W}} N_{\mathbf{K}}}(t)]$ is the vector form of the decision variable. $\mathbf{X}$ is the collection of $\mathbf{x}(t)$: $\mathbf{X} = [\mathbf{x}(1)^\top, \mathbf{x}(2)^\top, ..., \mathbf{x}(T-1)^\top]$. Decision time step refers to the specific time interval at which decision-making needs to be performed in the physical system.

The parameters a_{ijt}, o_{ijqt}, p_{ij}, $b_{ij}^{(1)}$, and $b_j^{(2)}$ are considered constant in this formulation, serving as instance parameters that determine deterministically the dynamics of the WTA. (1) is the objective function, and the problem is to find $\mathbf{X}$ that maximizes the probability of not being hit by a threat by the terminal timestep T. (2) is an stockpile constraint that ensures that the number of missiles W_i launches does not exceed P_i. (3) indicates that one missile can be launched per time. (4) defines the conditions under which threats cannot be intercepted. It reflects the operational limitations of an interceptor, specifically whether the interceptor can intercept the threat: $K_j \in \mathbf{A}_i$. It also establishes a bounded time window for threat interception, with a lower limit of $b_j^{(2)}$ and an upper limit of $b_{ij}^{(1)}$. In (5)-(6), the value of $y_{ij}(t,\tau)$, arrival-on-target indicator, is determined by the decision variable $x_{ij}(t)$. In (9)-(12), the number of missiles which are launched from interceptors in M_k and in the process of seizing simultaneous guidance resource does not exceed C_k. The simple example of this problem is described in supplementary material B.

III. Wargame Simulation Modeling

The mathematical model of (1)-(13) has the advantage of rigorously expressing constraints as a combinatorial optimization problem. However, its assumption that parameters a_{ijt}, o_{ijqt}, p_{ij}, $b_{ij}^{(1)}$, and $b_j^{(2)}$ are predetermined is unrealistic [15]. Furthermore, mathematical formulations are limited in their ability to represent the uncertain events that naturally occur in combat situations. Specifically, the uncertainties inherent in combat arise from multiple probabilistic factors: the stochastic nature of hit determination [27], probabilistic target acquisition [28], and interceptor error circular distributions [29]. To reflect these elements, we model the DWTA problem using the high-fidelity wargame simulation proposed by [30]. In addition, this framework incorporates RL API and Section II constraints to enable dynamic agent-environment interaction while preserving the combinatorial nature of the problem.

Fig. 1 depicts the stochasticity inherent in our wargame simulation. Surface-to-air missile (SAM) and close-in weapon system (CIWS) serve as the interceptor against missile-type threat and surface-to-surface missiles (SSM) can serve as an interceptor against launcher-type threat. ① In aiming position generation, the error radius of SAM increases quadratically with distance, and the aiming point is stochastically generated following a Gaussian distribution with target position as mean and error radius as variance. ② The target/decoy acquisition of SSM follows a multinomial distribution determined by radar cross sections (RCS) of targets and decoys. ③ The hit or

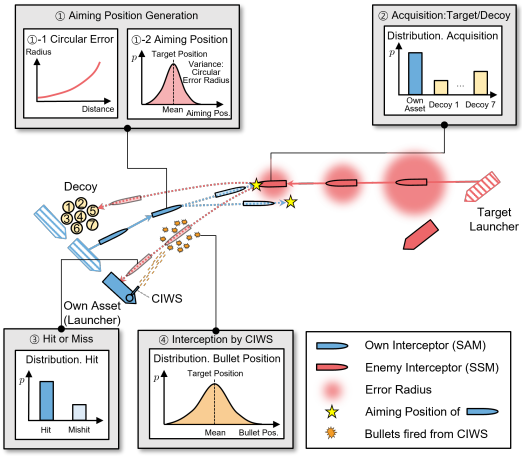


Fig. 1: Conceptual Diagram of Stochasticity in Our Wargame Simulation

miss distribution follows a binomial distribution, where the probability of each SSM/SAM's hit is stochastically determined. ④ In interception by CIWS, bullets are stochastically generated following a Gaussian distribution with target position as mean and variance as stated in [31], at a known rate of fire. The detailed simulation logic are described in supplementary material C.

IV. POMDP

To address both the operational uncertainty described in Section III and the partial observability arising from sensor detection limitations, we express DWTA as a POMDP, which models sequential decision-making under partially observable conditions. A POMDP consists of the following 7-tuple $< \mathcal{S}, \mathcal{U}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma >$.

- $\mathcal{S}$: Set of states
- $\mathcal{U}$: Set of actions
- $\mathcal{T}$: Transition function
- $\mathcal{R}$: Reward function
- Ω: Set of observations
- $\mathcal{O}$: Observation function
- γ: Discount factor: $\gamma \in [0, 1)$

In our study, the partial observability is caused by the limitation of the detection and the incomplete information about the enemy. The agent (the own asset), which interacts with the wargame simulation by performing an action, is the launcher that has several types of interceptors. Based on the observation on the environment, the agent takes an action according to the policy. The details about POMDP are covered in subsections.

A. Observation

The observation at timestep t, denoted as o_t is a observable, summarized and partial representation of the state s_t, and is a collection of these six features: $o_t = (f_1(t)||f_2(t)||f_3(t)||f_4(t)||f_5(t)||f_6(t))$, where $f_l(t)$ is the feature $l \in \{1, 2, 3, 4, 5, 6\}$. Detailed derivations and experimental analyses of each feature are provided in the supplementary material D.

a) Feature 1 : Feature 1 represents the cumulative number of missiles launched, and the feature 1 of W_i at timestep t, denoted as $f_{1,i}(t)$ is defined by (14)

$$f_{1,i}(t) = \sum_{\tau=1}^{t} \sum_{K_j \in \mathbf{K}} x_{ij}(\tau) \tag{14}$$

$f_1(t)$ is the collection of $f_{1,i}(t)$ for all $W_i \in \mathbf{W}$:$f_1(t) = [f_{1,i}(t)]_{i=1}^{N_\mathbf{W}}$. In experiment, each element is normalized by the initial number of each interceptor's missile.

b) Feature 2: Feature 2 represents the number of missiles in each interceptor sub-process. For q-th sub-process of W_i at timestep t, feature 2, denoted as $f_{2,i,q}(t)$, is defined in (15).

$$f_{2,i,q}(t) = \sum_{K_j \in \mathbf{A}_i} \sum_{h=1}^{t} w_{ijq}(t, h) \tag{15}$$

$f_{2,i}(t) = [f_{2,i,q}(t)]_{q=1}^{|\mathbf{E}_i|}$ and $f_2(t) = ||_{i=1}^{|\mathbf{W}|} f_{2,i}(t)$ denotes the concatenation of $f_{2,i}(t)$ across all $W_i \in \mathbf{W}$. In experiments, each element corresponding to W_i is normalized by its associated C_k such that $W_i \in \mathbf{M}_k$.

c) Feature 3: Feature 3 provides the distribution of threats located within the detection range of the own asset. The area within the detection range from the own asset can be divided into n_d segments where n_d refers to number of segments for discretizing the detection range. We define the set of threats in the e-th zone as

$$f_{3,e}(t) = |\{K_j \in \mathbf{K} | d\frac{e-1}{n_d} \leq d_j(t) \leq d\frac{e}{n_d}\}|, \quad e \in \mathbb{N}, 1 \leq e \leq n_d \tag{16}$$

where $d_j(t)$ represents the distance at timestep t from the own asset to threat K_j and d is the maximum detection range. $f_3(t) = [f_{3,e}(t)]_{e=1}^{n_d}$ repesents the collection of feature 3. In experiment, each element is normalized by the initial number of each enemy interceptor's missile.

d) Feature 4: Feature 4 quantifies the cumulative number of successfully intercepted threats up to time t. The feature 4 at timestep t, denoted as $f_4(t)$, consists of two elements: the cumulative number of successfully intercepted launcher-type threats $f_{4,1}(t)$ and missile-type threats $f_{4,2}(t)$: $f_4(t) = [f_{4,r}(t)]_{r=1}^{2}$.

e) Feature 5: Feature 5 represents the number of successful intercepts between time $t-1$ and t (myopic view), while feature 4 tracks total intercepts up to time t (long-term view). At timestep t, feature 5, denoted as $f_5(t)$, consists of two elements for launcher-type and missile-type threats: $f_5(t) = [f_{5,z}(t)]_{z=1}^{2}$, where $f_{5,1}(t)$ and $f_{5,2}(t)$ correspond to launcher-type and missile-type threats, respectively.

f) Feature 6: Feature 6 is the collection of threats' information that have been assigned interceptors from 1 step before to n_h steps before, where n_h is the time horizon for recording: $f_6(t) = [f_{6,y}(t)]_{y=1}^{n_h}$. $f_{6,y}(t)$ is the representation of the threat to which the interceptor was assigned y time step ago from t, and the representation of the target $f_{6,y}(t)$ includes the relative position and velocity with respect to the own asset (polar coordinate).

B. Action

The action u_t at timestep t is to choose one of the interceptable threats to attack at this time. The interceptability is related to the constraints in (2), (4), (12). The set of possible actions $\mathbf{U}(s_t = s)$ is the set of threats that includes the interceptable threats in state s: $\mathbf{U}(s_t = s) = \{K_j \in \mathbf{K} | \exists W_i; (K_j \in \mathbf{A}_i \wedge f_{1,i}(t) < P_i \wedge b_{ij}^{(1)} \leq t < b_j^{(2)}) \wedge (\sum_{W_i \in \mathbf{M}_k} f_{2,i}(t) < C_k; W_i \in \mathbf{M}_k)\} \cup K_0$ where K_0 is dummy threat indicating the own asset. If K_0 is chosen as an action, no interceptor will be launched against any threat: it means to do nothing. Also, K_0 is the only action that can be taken if there are no interceptable threat. The u_j denotes the action to launch a missile from the interceptor aimed at the threat K_j. The interceptor is selected by a distance-based rule (see supplementary material E), and a missile from that interceptor is launched and follows the specified sub-process.

1) Graph-based Action Representation: Graph-based action representation $\mathcal{G}(s) = (\mathcal{V}(s), (\mathcal{E}_1(s), \mathcal{E}_2(s), \mathcal{E}_3(s), \mathcal{E}_4(s), \mathcal{E}_5(s)))$ is a dynamic and heterogeneous graph constructed based on state s. $\mathcal{V}(s)$ is defined as the set of nodes corresponding to the own asset (friendly), threats (enemy), missiles launched by interceptor of the own asset as shown in (17). The nodes $v_n(s)$ are sorted in the order: dummy, interceptable, missiles of the interceptor (launched by the own asset), where dummy node refers to the own asset. A node $v_n(s)$ with $n \geq |\mathbf{U}(s)|$ corresponds to missiles of interceptor.

$$\mathcal{V}(s) = \{ \underbrace{v_0(s)}_{\substack{\text{dummy threat} \\ \text{(own asset)}}}, \underbrace{v_1(s), ..., v_{|\mathbf{U}(s)|-1}(s)}_{\text{interceptable threats}}, \underbrace{v_{|\mathbf{U}(s)|}(s), ..., v_{|\mathbf{U}(s)|+|\mathbf{L}(s)|-1}(s)\}}_{\text{missiles of interceptor}} \tag{17}$$

where $\mathbf{L}(s)$ is the number of interceptor missiles which are launched by the own asset and flying in the skies.

- $\mathcal{E}_1(s)$ represents bidirectional connections between the own assets and threats that fall within the detection range of the assets.
- $\mathcal{E}_2(s)$ is bidirectional links between threats within distance (d_a). The proximity threshold d_a is a user-defined hyperparameter.
- $\mathcal{E}_3(s)$ is bidirectional links between the threat and the missile of interceptor. If $v_n(s); n \geq |\mathbf{U}(s)|$ corresponding to the missile of the interceptor flying towards the threat $v_n(s); 1 \leq n \leq |\mathbf{U}(s)| - 1$, they are connected, otherwise disconnected.
- $\mathcal{E}_4(s)$ is a bidirectional links between launcher-type threats and their launching missile-type threats.
- $\mathcal{E}_5(s)$ is bidirectional links between flying missiles of interceptors and the own asset.

A node $v_n(s)$ has a feature vector $z_n(s) = (\mathfrak{p}_r(s; v_n), \mathfrak{p}_\theta(s; v_n), \mathfrak{u}_r(s; v_n), \mathfrak{u}_\theta(s; v_n))$, where each component of $z_n(s)$ represents the relative position ($\mathfrak{p}_\cdot(\cdot)$) and relative velocity ($\mathfrak{u}_\cdot(\cdot)$) with respect to the reference node in polar coordinate, respectively. The reference node for dummy node $v_0(s)$ and threat node $v_n(s); n > |\mathbf{U}(s)| - 1$ is the own asset itself (i.e., $z_0(s) = (0, 0, 0, 0)\ \forall s \in \mathbf{S}$) and for missile of intercepting $v_n(s); 1 \leq n \leq |\mathbf{U}(s)| - 1$ is the targeting threat node which each missile is flying toward.

C. Reward

To align the cumulative sum with the true objective function and address the sparse reward problem, we define a shaped reward $r'_t(H)$ as shown in (18).

$$r'_t(H) = Hr_{1,t} + r_{2,t} \tag{18}$$

where $r_{1,t} = 1$ if all threats being successfully intercepted (otherwise 0) and $r_{2,t}$ is the number of threats intercepted from t to $t+1$ and H is a non-negative constant.

Proposition 1. *Assuming $\sum_{t=1}^{T} r_{2,t}$ is bounded, as H goes to infinity ($H \to \infty$), $J(\pi)$ and $J(\pi)'$ have a strict monotonic increasing relationship.*

Proof. $0 \leq \sum_{t=1}^{T} r_{2,t} \leq c$ where c is the upper bound. $\mathbb{E}_{\tau\sim\pi}[\sum_{t=1}^{T} r'_t(H)] \leq \max \sum_{t=1}^{T} r'_t(H) = \mathbb{E}_{Y=1}[\sum_{t=1}^{T} r'_t(H)] = H + c$

$$\begin{aligned} J'_H(\pi) &= \mathbb{E}_{\tau\sim\pi}[\sum_{t=1}^{T} r'_t(H)] \\ &= \sum_{y\in\{0,1\}} P(Y = y|\tau \sim \pi)\mathbb{E}_{Y=y}[\sum_{t=1}^{T} r'_t(H)] \\ &= P(Y = 1|\tau \sim \pi)(H + c) + \\ &\quad P(Y = 0|\tau \sim \pi)\mathbb{E}_{Y=0}[\sum_{t=1}^{T} r'_t(H)] \\ &= (H + c)J(\pi) + (1 - J(\pi))C(\pi) \end{aligned} \tag{19}$$

where $C(\pi) := \mathbb{E}_{Y=0}[\sum_{t=1}^{T} r'_t(H)]$.
Given two policies π_1 and $\pi_2 (J'_H(\pi_1) < J'_H(\pi_2))$,

$$\begin{aligned} J'_H(\pi_2) - J'_H(\pi_1) = (H + c)[J(\pi_2) - J(\pi_1)] + C(\pi_1)J(\pi_1) - \\ C(\pi_2)J(\pi_2) - C(\pi_1) + C(\pi_2) > 0 \end{aligned}$$

therefore, $J'_H(\pi_1) < J'_H(\pi_2)$ implies $J(\pi_1) < J(\pi_2)$ □

where $J(\pi)$ and $J'(\pi)$ represent the true and shaped objective functions under policy π, respectively. The shaped objective function $J'(\pi)$ is defined as the expected cumulative sum of shaped rewards: $J'(\pi) := \mathbb{E}_{\tau\sim\pi}[\sum_{t=1}^{T} r'_t(H)]$. According to Proposition 1, there is a strict monotonic increasing relation between the original objective function and the shaped objective function. For numerical stability, the value of H is appropriately adjusted as a hyperparameter. The detailed derivation is described in supplementary material F.

V. Neural Network Architecture

The proposed decision-making neural network (NN) model consists of three parts: observation embedding, action embedding, and policy network. Fig. 2 illustrates the network architecture and information flow of the proposed framework.

A. Observation Embedding

An observation o comprising six features is embedded into a higher dimension using a function $\phi(\cdot)$ to improve its representation. $\phi(o)$ facilitates state estimation under

the partial observability conditions. The observation embedding function $\phi(\cdot)$ is implemented using a multi-layered perceptron (MLP). Batch normalization is applied, and the output dimension is d_o.

B. Action Embedding

We adopt the GNN architecture, graph attention networks (GAT) [32] for action embedding. The feature vector of v_n, denoted as $z_n(s)$, is fed into this GNN architecture and thus serves as the initial (0-th layer) action embedding: $h_n^{(0)}(s) = z_n(s)$. The kth layer action embedding function $\mathcal{F}_i^{(k)}$ for the edge type $\mathcal{E}_i(\cdot)$ forms the kth action embedding, as shown in the 20.

$$h_{i,n}^{(k)}(s) = \mathcal{F}_i^{(k)}(h_n^{(k-1)}(s), \mathcal{N}(v_n(s), \mathcal{E}_i(s))) \quad (20)$$

$$= \sigma\Big(\sum_{\substack{v_m(s) \in \\ \mathcal{N}(v_n(s), \mathcal{E}_i(s))}} \alpha_{nm}^{i,(k)}(s) \mathcal{W}_V^{i,(k)} h_m^{(k-1)}(s) \Big) \quad (21)$$

where $\mathcal{N}(v, \mathcal{E})$ is the set of nodes connected to node v by edge $\mathcal{E}$ and $\mathcal{W}_V^{i,(k)}$ is a learnable parameter. For $k \neq 1$, $\mathcal{W}_V^{i,(k)}$ has dimensions $d_V \times d_V$, while for $k = 1$, it has dimensions $d_V \times 4$. Here, d_V represents the dimension of $h_{i,\cdot}^{(k)}(s)$. The attention coefficient $\alpha_{nm}^{i,(k)}(s)$ is defined as 22.

$$\alpha_{nm}^{i,(k)}(s) = \frac{\exp(\psi(\alpha_{\text{attn}}^{i,(k)} \cdot \mathcal{W}_Q^{i,(k)} h_n^{(k-1)}(s) || \mathcal{W}_K^{i,(k)} h_m^{(k-1)}(s)))}{\sum_{\substack{v_{m'} \in \\ \mathcal{N}(v_n, \mathcal{E}_i(s))}} \exp(\psi(\alpha_{\text{attn}}^{i,(k)} \cdot \mathcal{W}_Q^{i,(k)} h_n^{k-1}(s) || \mathcal{W}_K^{i,(k)} h_{m'}^{(k-1)}(s)))} \quad (22)$$

where ψ is a LeakyReLU activation function and $\mathcal{W}_Q^{i,(k)}$, $\mathcal{W}_K^{i,(k)}$, $\alpha_{\text{attn}}^{i,(k)}$ are learnable parameters. For $k \neq 1$, $\mathcal{W}_Q^{i,(k)}$ and $\mathcal{W}_V^{i,(k)}$ have dimensions $d_{\text{attn}} \times d_V$, respectively, while for $k = 1$, they have dimensions $d_{\text{attn}} \times 4$. $\alpha_{\text{attn}}^{i,(k)}$ has dimension $2 \times d_{\text{attn}}$. By using attention mechanism, the function aggregates the information of the neighboring nodes connected by each edge. It outputs a new embedding that reflects the topological structure. The embeddings for each edge types are concatenated to form the kth embedding, as shown in (23).

$$h_n^{(k)}(s) = \mathcal{H}^{(k)}(h_{1,n}^{(k)}(s) || h_{2,n}^{(k)}(s) || h_{3,n}^{(k)}(s) || h_{4,n}^{(k)}(s) || h_{5,n}^{(k)}(s)) \quad (23)$$

where $||$ is concatenating operator and $\mathcal{H}^{(k)}$ is MLP that reduces the dimension of input vector into d_V. By repeatedly applying (20)-(23), the information of neighboring nodes beyond 1-hop can be blended in the embedding. That is, $h_n^{(k)}(s)$ is a k-hop action embedding for $v_n(s) \in \mathcal{V}(s)$, where $k \geq 1$, and $h_n^{(k)}(s)$ utilizes information from neighboring nodes that are k-hops away for embedding. If K layers are stacked in total for the action embedding, the output $h_n^{(K)}(s)$ from the final layer of the GNN is delivered to the policy network as stated in the subsequent section. $v_n(s)$ is uniquely assigned to each threat, where $n \leq |\mathbf{U}(s)| - 1$. Therefore, $h_n^{(K)}(s)$ refers to the action embedding corresponding to each action u_n attacking the threat K_n, where $u_n \rightarrow h_n^K(s)$. $h^K(s)$ denotes the collection of action embeddings for all possible actions: $h^{(K)}(s) = (h_0^{(K)}(s), ..., h_{|\mathbf{U}(s)|-1}^{(K)}(s))$.

C. Policy Network

At this point, we have the representation of state s and action u, and they are $\phi(o)$ and $h^K(s)$, respectively. Policy π takes state s as input and outputs the probability of action u (i.e., $\pi(u|s)$ in general policy form). In our case, this can be interpreted as a probability distribution that takes o and $h^K(s)$ as input and outputs the probability of choosing the action corresponding to each node (threat). Specifically, for scalability regardless of the number of threats, the policy network takes $\phi(o)||h_n^K(s)$ as input for each action $u_n \in \mathbf{U}(s)$ and outputs a scalar value for the action. These scalar values are passed through a softmax function as shown in (24) to obtain the probability of selecting u_n.

$$\pi(u_n|s) \approx \frac{\exp(MLP(\phi(o)||h_n^K(s)))}{\sum_{m=0}^{|\mathbf{U}(s)|-1} \exp(MLP(\phi(o)||h_m^K(s)))} \quad (24)$$

The agent makes decisions randomly by choosing an action sampled from the policy distribution $[\pi(u_n|s)]_{n=0}^{|\mathbf{U}(s)|-1}$.

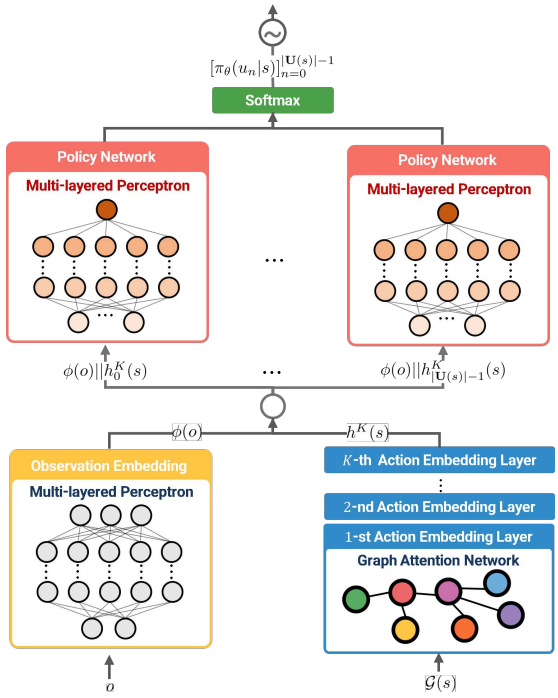


Fig. 2: NN Architecture

VI. Policy Optimization

Proximal policy optimization (PPO) [33] is used to optimize the NN architecture. PPO uses clipped surrogate objective function to improve its policy. The aim of this clipping method is to limit excessive policy updates by suitably adjusting the step size in the policy space. PPO also improves sample efficiency by utilizing previous experience (old sample) for training. The clipped surrogate objective function $L_{CLIP}(\theta)$ in PPO is defined as:

$$L_{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta))\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (25)$$

where θ is the collection of learnable parameters for the observation embedding, action embedding, and policy network. Here, $r_t(\theta)$ denotes the probability ratio $\pi_\theta/\pi_{\theta_{old}}$ and ϵ is a clipping parameter. Old policy $\pi_{\theta_{old}}$ denotes the policy from which trajectories are sampled before being updated. To estimate advantage $\hat{A}_t$, we use the generalized advantage estimator (GAE) [34], which is computed as in (26).

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \tag{26}$$

where the temporal-difference (TD) error δ_t is defined as $r_t + v(s_{t+1}) - v(s_t)$. To approximate $v(s)$, $\phi(o)$ is fed into the critic network, which shares parameters with the policy network except for the input and output layers. The learning process is a policy optimization process that adjusts the parameter θ to maximize L_{CLIP}. The optimization process iterates for N_{epochs} epochs using the old samples.

Algorithm 1 outlines the policy optimization process. For $N_{episodes}$ iterations, trajectories are collected from randomly generated wargame instances using the current policy. Each trajectory consists of observations o, graph-based action representation $\mathcal{G}(s)$, actions u, and rewards r: $(o_t, \mathcal{G}(s_t), u_t, r_t)_{t=1}^T$. The network parameters are updated for N_{epochs} iterations, where the optimization minimizes L_{Total} consisting of L_{CLIP} and L_{Critic}.

Algorithm 1 Policy Optimization Procedure

Initialize parameters θ of actor network and ω of critic network
while not converged **do**
  trajectories $\leftarrow \emptyset$
  for $i = 1$ to $N_{episodes}$ **do**
    Sample problem instance from combat scenario distribution and initialize wargame simulation
    Generate trajectory $(o_t, \mathcal{G}(s_t), u_t, r_t)_{t=1}^T$ by executing agent policy in the simulation
    trajectories $\leftarrow$ trajectories $\cup$ $(o_t, \mathcal{G}(s_t), u_t, r_t)_{t=1}^T$
  end for
  for epoch $= 1$ to N_{epochs} **do**
    for each batch in trajectories **do**
      $\phi(o) \leftarrow$ ObservationEmbedding(o)
      $h^{(K)}(s) \leftarrow$ ActionEmbedding($\mathcal{G}(s)$)
      $\pi(u|s) \leftarrow$ PolicyNetwork($\phi(o), h^{(K)}(s)$)
      $v(s) \leftarrow$ CriticNetwork($\phi(o)$)
      $\hat{A} \leftarrow$ ComputeGAE(v, γ, λ)
      $L_{CLIP} \leftarrow$ ComputePPOLoss($\pi, \pi_{old}, \hat{A}$)
      $L_{Critic} \leftarrow$ ComputeMSELoss(v, r)
      $L_{Total} \leftarrow -L_{CLIP} + 0.5L_{Critic}$
      Update parameters to minimize L_{Total}
    end for
  end for
end while

VII. Experiment

In this study, we validate the effectiveness of the proposed method through a series of experiments involving both ground and naval combat engagements. In ground combat scenarios, the experiments simulate missile engagements between transporter erector launchers (TELs), while naval combat scenarios examine missile exchanges between ships. The performance is measured by the probability of complete survival (PCS), defined as the ratio of trials with zero hits over total trials, which represents the likelihood that all incoming missiles are successfully intercepted, aligning with the objective function (1). To prevent interference between friendly interceptor missiles, the decision-making timestep in the experiment is set to 4 seconds. The experimental datasets and detailed specifications for TEL and ship platforms are included in the supplementary material G.

TABLE I: Hyperparameters

Parameter	Value	Parameter	Value
d_o	52	γ	0.99
d_{attn}	72	λ	0.95
d_V	42	ϵ	0.18
n_d	12	N_{epochs}	2
n_h	10	$N_{episodes}$	35
d_a	10	Learning Rate	0.88e-4
Optimizer	Adam	-	-

TABLE II: Network Architecture of MLPs

Layer	Hidden Units
Observation Embedding	[128, 64, 48, 39, 32]
Policy Network	[126, 108, 64]
$\mathcal{H}^k$	[d_V, 23]
Activation Function	ELU

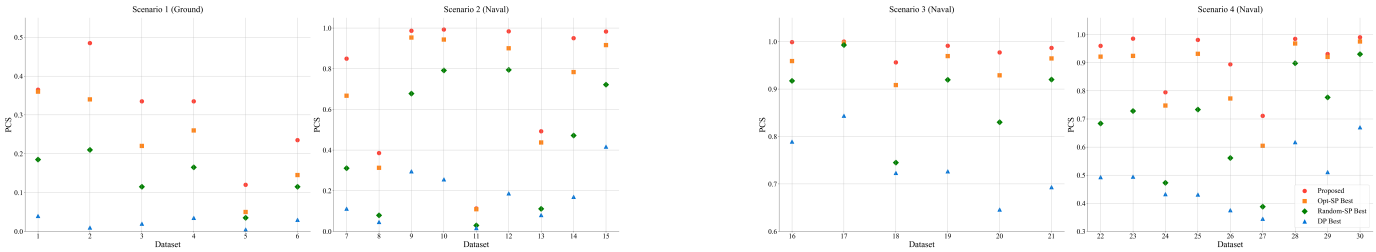
A. Training

In training, a type A friendly TEL/ship engages three enemy TELs/ships (types B, C, and D). The proposed method is applied to the friendly unit's decision-making, with two models trained for ground and naval combat scenarios. The encounter distance and angle between friendly and enemy units are randomized each episode[2]. After random hyperparameter search (detailed in Tables I and II), the ground combat model was trained for 46.67 hours (33,110 episodes) and the naval combat model for 33.28 hours (15,450 episodes).

B. Test

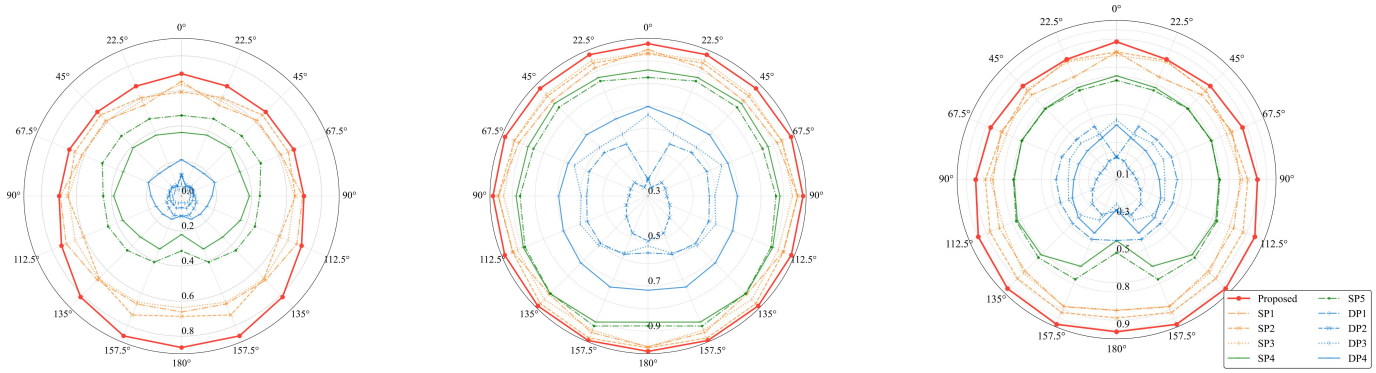
In the test phase, the enemy group engages friendlies using stochastic policy (SP) 0, consistent with the training phase (Details for SP 0 are described in the supplementary

[2]Initial encounter distances follow $\mathcal{N}(90, 3^2)$ for ground combat and $\mathcal{N}(60, 3^2)$ for naval combat. Encounter angles follow $\mathcal{U}(0, 360)$ for ground combat and $\mathcal{U}(65, 115)$ for naval combat.



(a) Generalization Test (b) Scalability Test

Fig. 3: Performance Analysis Results((a) Generalization Test, (b) Scalability Test)



(a) Scenario 2 (b) Scenario 3 (c) Scenario 4

Fig. 4: PCS by Encounter Angle in Generalization and Scalability Test ((a) scenario 2, (b) scenario 3, (c) scenario 4)

material C.3). For comparison, we adopt 9 additional baselines: 4 deterministic policies (DPs) and 5 SPs. DPs are rule-based approaches that reflect deterministic preferences such as threat distance or speed. SPs are stochastic approaches based on threat distance, speed, engagement range, or purely random selection. Among the five SPs, SP1-3 use GA optimization (denoted as Opt-SP), while SP4-5 use uniformly random policies (denoted as Random-SP). Opt-SPs combine GA-optimized performance with inherent unpredictability of human decision-making . Details about the baselines are provided in the supplementary material H.

We conduct several tests to evaluate the generalization ability and scalability of the proposed method. In Section VII-B1, the two models are tested in situations involving a single friendly TEL or ship against multiple enemy TELs or ships, respectively. In Section VII-B2, we verify that the proposed methodology works effectively in multi-agent situations where the number of friendly ships varies.

1) Generalization Test: We evaluate the generalization ability by testing one friendly TEL/ship against multiple enemies across two scenarios: ground (scenario 1) and naval combat (scenario 2). A total of 15 datasets are used (dataset 1-6 for scenario 1 and dataset 7-15 for scenario 2). For scenario 1, each dataset is tested with 200 episodes at a single encounter angle. For scenario 2, each dataset is tested with 200 episodes at 9 different encounter angles (0° to 180° in 22.5° increments, with episodes randomly generating orientations and positions within the specified direction ranges), resulting in 1,800 test episodes per dataset. Fig. 3a illustrates PCS across datasets. The proposed method consistently outperforms other approaches, demonstrating superior generalization across untrained scenarios. Fig. 4a demonstrates that the proposed method achieves the highest performance and lowest performance fluctuation compared to other baselines across all encounter angles. This demonstrates that the proposed method maintains its robustness regardless of factors such as detection probability variations due to RCS changes from different encounter angles and node feature variations in 2D space.

2) Scalability test: The scalability test involves naval combat scenarios with multiple friendly and enemy ships, using 15 additional datasets. Scenario 3 (dataset 16-21) involves combat with a relatively lower ratio of enemy ships, while Scenario 4 (dataset 22-30) involves combat with a relatively higher ratio of enemy ships. For each dataset, 200 episodes are tested per encounter angle (0° to 180° in 22.5° increments, with the randomness as in generalization test). Fig. 3b shows the proposed method's PCS, consistently outperforming baseline methods across all datasets.

Fig. VII-B2 shows PCS by encounter angle in Scenarios 3 and 4. The proposed method consistently outperforms other methods across all encounter angles in both scenarios, indicating robust performance regardless of the relative positioning of ships. Notably, the proposed method consistently maintains superior performance around 0.8-1.0 across all encounter angles, while SP1-SP3 show generally good performance but with subtle fluctuations. SP4-

SP5 demonstrate moderate performance around 0.6, and DP1-DP4 show relatively poor performance below 0.4 across all angles.

Table III illustrates the average PCS across different scenarios. Notably, in the generalization test (Scenarios 1 and 2), the difference between the proposed method and baseline is 0.08, whereas in the scalability test (Scenarios 3 and 4), this difference reduces to 0.04 and 0.05, respectively. Based on these results, we observed meaningful findings regarding the method's performance: the proposed approach demonstrates potential scalability in multiple unit scenarios while revealing the inherent complexity of decision-making in multi-unit environments.

	Proposed	**Opt-SP Best**	**Random-SP Best**	**DP Best**
Scenario1	0.31	0.23	0.14	0.02
Scenario2	0.75	0.67	0.44	0.11
Scenario3	0.99	0.95	0.89	0.67
Scenario4	0.91	0.86	0.69	0.47

TABLE III: Average PCS for scenarios

C. Comparisons with other DRL Approaches

To validate the performance of the proposed method, we compare it with actor critic from [17], DQN from [24], [16], and recent DQN variants (implicit quantile networks [35], priority replay buffer [36], dueling DQN [37], n-step TD). For each method, we train with 5 different hyperparameter sets for the naval combat training scenario and record the average PCS for 20 validation episodes on dataset 7 every 10 episodes (see supplementary material I.1). As the problem settings differ across studies, we train the baseline methods under our experimental conditions. Fig. 5 shows the learning curves of each DRL approaches. The proposed method achieved over 0.7 PCS after 6,000 epochs, while actor critic converged to 0.4 PCS. DQN and its variants showed no policy improvement, remaining at the initial performance level of 0.1. The solid lines indicate mean values and shaded regions represent 25-75 percentile ranges across different hyperparameter sets.

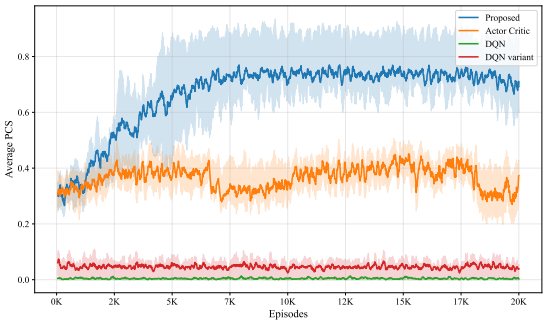


Fig. 5: Learning Curve of DRL approaches

D. Analysis of GNN

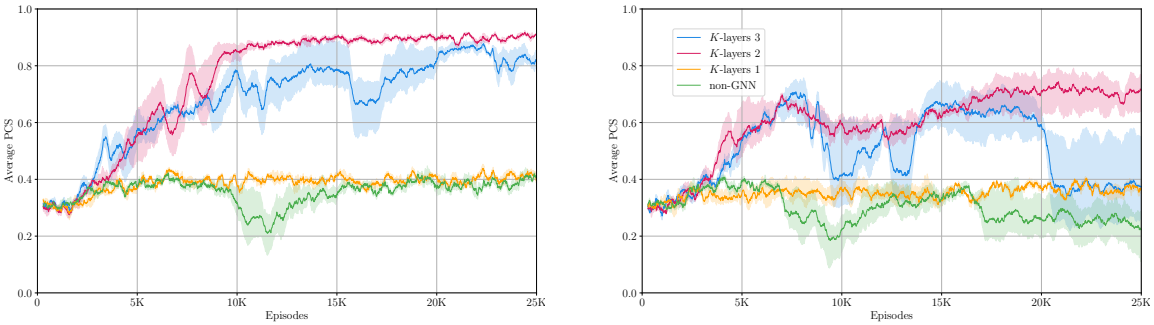
1) Impact Analysis of GNN Layer Depth: This study evaluates GNN performance by comparing 1-, 2-, and 3-layer structures against a non-GNN model. For each structure, we select the top three and bottom three performing hyperparameter combinations from six training runs in naval combat scenarios (see supplementary material I.2). Training runs for 25,000 epochs, with PCS measured on validation set (dataset 7) every 10 epochs for 20 epochs. The visualization shows the average performance (solid line) and 25-75 percentile range (transparent color) for each K-layer structure. Fig. 6 shows the results.

Results show all GNN structures consistently outperform the non-GNN model. The 2-layer and 3-layer GNNs show steady improvement in both worst-case and 25th percentile performance, achieving higher PCS, while the non-GNN model remains poor. The 2-layer and 3-layer structures demonstrate clear advantages, with their best cases significantly exceeding the non-GNN model's performance. Notable performance differences exist between the 2-layer and 3-layer structures:

- Overall Performance: The 2-layer structure consistently outperforms the 3-layer structure throughout the learning process in both best and worst cases.
- Learning Stability: The 2-layer structure showed a narrower performance range (25-75 percentile), indicating more stable learning, while the 3-layer structure exhibited wider performance variation and greater instability, particularly during the 10K-15K epoch interval.

2) t-SNE Visualization of GNN Representations: The colored scatter plot in the center of Fig. 7 shows the t-SNE (t-distributed stochastic neighbor embedding) visualization of node features from the layer preceding the policy network's output, reduced to two dimensions. Each node represents observation-selected action pairs from 500 test runs, colored by TD target value ($r_t + \gamma v(s_{t+1})$). This visualization reveals clustering and distribution patterns of states according to the TD target.

The central visualization is surrounded by histograms representing the values of observation feature 3 for each point. We focused on analyzing the visualization in terms of observation feature 3 because feature 3 intuitively and perceptually illustrates the concept of current threat level. In Fig. 7, we observe that states with similar threat levels are positioned close to each other in the embedding space, maintaining perceptual similarity as spatial proximity. This is evident in pairs 1-2, 3-4, and 7-8. Conversely, pairs 5-6, 9-10, and 11-12 are positioned close together in the embedding space despite having relatively low similarity in the detailed aspects of their distributions. Additionally, we note pairs like 5 and 9, or 1 and 4, where threat distributions are similar but are positioned far apart in the embedded space. These observations suggest that the graph embedding through the GNN captures more complex representations beyond simple perceptual similarity.



(a) Best 3 hyperparameters (b) Worst 3 hyperparameters

Fig. 6: Learning Curves according to K-layers

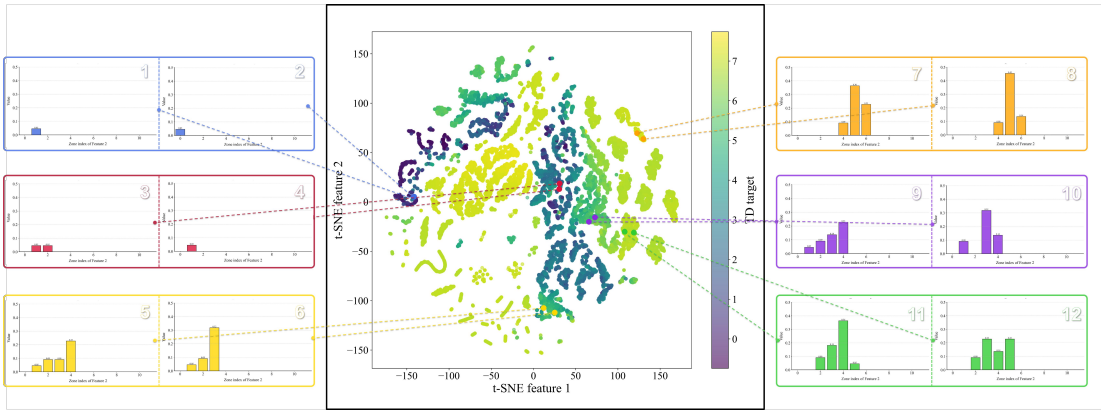


Fig. 7: t-SNE Visualization of Hidden Features of Last Layer in Policy Network

E. Analysis of the Proposed Decision-making Strategy

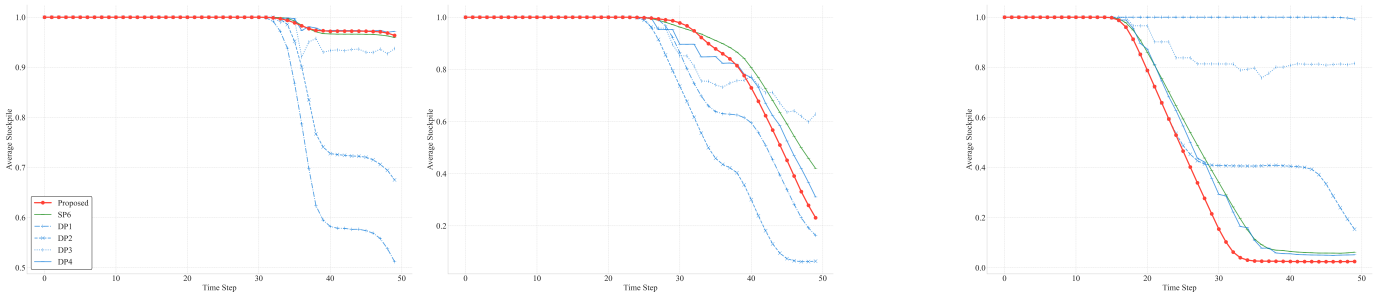
To analyze the proposed decision-making strategy, 1,000 test episodes are conducted for the ground battle scenario (dataset 1). Fig. 8 shows the status of stockpiles (i.e., remaining quantity) changes of each interceptor over timesteps, while Fig. 9 records the count of threats that hit based on their distance from the own asset. Long-range and middle-range interceptors correspond to SAM, while short-range corresponds to CIWS. Analyzing the proposed method and comparing with other methods reveals strategic engagement patterns based on threat distances and accuracy considerations. The method aggressively depletes long-range interceptors ($\leq$60NM) by timestep 30 (Fig. 8(c)), compensating for their lower accuracy due to long distance. While interception attempts decrease sharply beyond 40NM, the engagement activity shows a distinct intense concentration in a narrow band around 25-30NM, as red in Fig. 9. These nuanced patterns of the proposed method show a distinct difference from those that are overly concentrated at specific distance ranges (DP1, DP2) or widely distributed (SP4, DP3, DP4). Medium-range interceptors ($\leq$30NM) show 75% usage (Fig. 8(b)), while highly accurate short-range interceptors ($\leq$4NM) maintain 95% of their stockpile (Fig. 8(a)). This suggests that the proposed method takes into account the fact that despite the high accuracy of short-range interception, engaging at this range should be avoided if possible, as it serves as a critical last line of defense.

This coordinated pattern between stockpile consumption and interception distances reveals a deliberate layered defense strategy: intense early engagement of distant threats using larger quantities of long-range interceptors to overcome accuracy degradation at distant range, while avoiding wasteful attempts at extreme distances. This is followed by concentrated medium-range interceptions, while preserving highly accurate short-range interceptors for critical close-range defense. Unlike other methods which show more scattered (SP4, DP3, DP4) or biased (DP1, DP2) interception patterns, this approach demonstrates a systematic transition from long to short-range engagements, adapting interceptor usage based on the inherent accuracy-range tradeoff.

VIII. Discussion

The proposed method demonstrates high generalization and scalability due to the following characteristics:

- **Representation of Topological Relationships:** GNN architecture effectively captures crucial information for quality decision-making embedded in graph-based action representation (Section VII-D2), leading to significant performance gains (Section VII-C, VII-D1). While Section VII-D1 shows the GNN approach significantly outperforms baseline non-GNN methods, the analysis reveals optimal performance



(a) (b) (c)

Fig. 8: Interceptor Stockpile Status over Time for: (a) Short-range Interceptor (≤4NM), (b) Medium-range Interceptor (≤30NM), and (c) Long-range Interceptor (≤ 60NM)

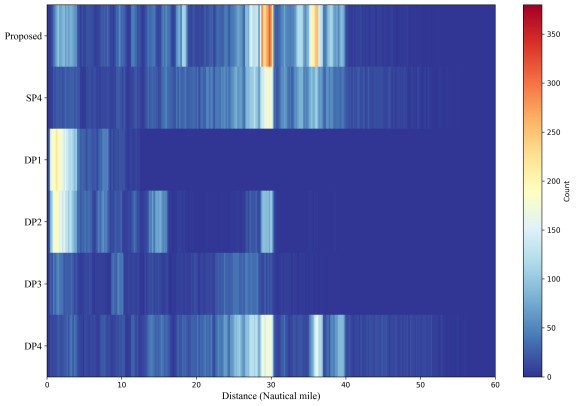


Fig. 9: Number of Intercepted Threats by Distance

with moderate network depth (2-layer) due to over-smoothing effects [3] in deeper architectures.

- **Scalability based on Graph Approach:** In addition to the rerpesentation improvement, GNN and graph-based action representation enhances scalability by just utilizing local information from neighboring nodes. This local approach achieves better efficiency compared to global approaches that suffer from information overflow and redundancy [38].
- **Informative observation features:** Specifically, these observation features derived from dynamics and objective function perspectives provide crucial information for estimating the value function, contributing significantly to reaching a good policy.
- **Sophisticated interception strategy:** The proposed method considers interceptor stockpile status and distance-dependent hit probabilities, yielding a sophisticated engagement strategy that delivers superior performance.

IX. Conclusion

In this study, we proposed a GNN-DRL based combat decision-making model for optimizing survivability. We developed the model using a high-fidelity wargame simulation environment and introduced key components including observation features, graph-based action representation, and reward for DWTA. Our NN architecture, comprising observation embedding, action embedding, and policy network are optimized by using PPO. Experimental results demonstrated superior generalization and scalability in multi-domain environments compared to baseline methods. Furthermore, we confirmed that the GNN implementation significantly contributed to performance improvement.

As limitations, our DRL approach lacks generalization performance guarantees and does not address multi-agent cooperation. Future work will integrate hyperheuristic-based DRL [39] and explore cooperative multi-agent reinforcement learning frameworks [40], [37].

[3]Deeper architectures exhibit oversmoothing, where excessive message passing homogenizes node features and reduces model performance.

References

[1] S. P. Lloyd and H. S. Witsenhausen, "Weapons allocation is np-complete," in *1986 summer computer simulation conference*, Conference Proceedings, pp. 1054–1058, doi: https://doi.org/10.1007/springerreference_5892.

[2] B. Xin, Y. Wang, and J. Chen, "An efficient marginal-return-based constructive heuristic to solve the sensor–weapon–target assignment problem," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2536–2547, 2019.

[3] D. E. Blodgett, M. Gendreau, F. Guertin, J.-Y. Potvin, and R. Séguin, "A tabu search heuristic for resource management in naval warfare," *Journal of Heuristics*, vol. 9, no. 2, pp. 145–169, 2003. [Online]. Available: https://doi.org/10.1023/A:1022525529778

[4] S. A. Burr, J. E. Falk, and A. F. Karr, "Integer prim-read solutions to a class of target defense problems," *Operations Research*, vol. 33, no. 4, pp. 726–745, 1985, doi: https://doi.org/10.1287/opre.33.4.726.

[5] R. A. Murphey, *An Approximate Algorithm For A Weapon Target Assignment Stochastic Program.* Boston, MA: Springer US, 2000, pp. 406–421. [Online]. Available: https://doi.org/10.1007/978-1-4757-3145-3_24

[6] S.-c. Chang, R. M. James, and J. J. Shaw, "Assignment algorithm for kinetic energy weapons in boost phase defence," in *26th IEEE conference on decision and control*, vol. 26. IEEE, Conference Proceedings, pp. 1678–1683, doi: https://doi.org/10.1109/cdc.1987.272755.

[7] R. M. Soland, "Optimal terminal defense tactics when several sequential engagements are possible," *Operations Research*, vol. 35, no. 4, pp. 537–542, 1987, doi: https://doi.org/10.1287/opre.35.4.537.

[8] O. Karasakal, "Air defense missile-target allocation models for a naval task group," *Computers & Operations Research*, vol. 35, no. 6, pp. 1759–1770, 2008, part Special Issue: OR Applications in the Military and in Counter-Terrorism. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030505480600222X

[9] L. Wu, H.-y. Wang, F.-x. Lu, and P. Jia, "An anytime algorithm based on modified ga for dynamic weapon-target allocation problem," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE,

Conference Proceedings, pp. 2020–2025, doi: https://doi.org/10.1109/cec.2008.4631065.

[10] C. Leboucher, H.-S. Shin, P. Siarry, R. Chelouah, S. Le Ménec, and A. Tsourdos, "A two-step optimisation method for dynamic weapon target assignment problem," *Recent advances on meta-heuristics and their application to real scenarios*, pp. 109–129, 2013, doi: https://doi.org/10.5772/53606.

[11] P. A. Hosein and M. Athans, "The dynamic weapon-target assignment problem," *Technical Report LIDS-TH-1922*, 1989, doi: https://doi.org/10.1109/wsc.2013.6721653.

[12] K. Zhang, D. Zhou, Z. Yang, X. Li, Y. Zhao, and W. Kong, "A dynamic weapon target assignment based on receding horizon strategy by heuristic algorithm," in *Journal of Physics: Conference Series*, vol. 1651. IOP Publishing, Conference Proceedings, p. 012062, doi: https://doi.org/10.1088/1742-6596/1651/1/012062.

[13] B. Xin, Y. Wang, and J. Chen, "An efficient marginal-return-based constructive heuristic to solve the sensor–weapon–target assignment problem," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2536–2547, 2018, doi: https://doi.org/10.1109/tsmc.2017.2784187.

[14] J. Li, G. Wu, and L. Wang, "A comprehensive survey of weapon target assignment problem: Model, algorithm, and application," *Engineering Applications of Artificial Intelligence*, vol. 137, p. 109212, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197624013708

[15] D. P. Bertsekas, M. L. Homer, D. A. Logan, S. D. Patek, and N. R. Sandell, "Missile defense and interceptor allocation by neuro-dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 1, pp. 42–51, 2000, doi: https://doi.org/10.1109/3468.823480.

[16] W. Luo, J. Lü, K. Liu, and L. Chen, "Learning-based policy optimization for adversarial missile-target assignment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4426–4437, 2022.

[17] A. Dabholkar, J. Z. Hare, M. Mittrick, J. Richardson, N. Waytowich, P. Narayanan, and S. Bagchi, "Adversarial attacks on reinforcement learning agents for command and control," *The Journal of Defense Modeling and Simulation*, p. 15485129241271178, 2024, doi: https://doi.org/10.1177/15485129241271178.

[18] H. Huang, Z. Hu, Z. Lu, and X. Wen, "Network-scale traffic signal control via multiagent reinforcement learning with deep spatiotemporal attentive network," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 262–274, 2023.

[19] W. Hu, F. Chen, L. Xiang, and G. Chen, "Multi-asv coordinated tracking with unknown dynamics and input underactuation via model-reference reinforcement learning control," *IEEE Transactions on Cybernetics*, vol. 53, no. 10, pp. 6588–6597, 2023, doi: https://doi.org/10.1109/tcyb.2022.3203507.

[20] Q. Qu, L. Geng, K. Liu, and J. Lü, "Learning-based reconfiguration of charged spacecraft formation in geomagnetic field," *IEEE Transactions on Cybernetics*, pp. 1–12, 2024.

[21] X. Li, D. Zhou, Q. Pan, Y. Tang, and J. Huang, "Weapon-target assignment problem by multiobjective evolutionary algorithm based on decomposition," *Complexity*, vol. 2018, no. 1, p. 8623051, 2018, doi: https://doi.org/10.1155/2018/8623051.

[22] J. Huang, X. Li, Z. Yang, W. Kong, Y. Zhao, and D. Zhou, "A novel elitism co-evolutionary algorithm for antagonistic weapon-target assignment," *IEEE Access*, vol. 9, pp. 139 668–139 684, 2021, doi: https://doi.org/10.1109/access.2021.3119363.

[23] J. M. Rosenberger, H. S. Hwang, R. P. Pallerla, A. Yucel, R. L. Wilson, and E. G. Brungardt, "The generalized weapon target assignment problem," in *10th International Command and Control Research and Technology Symposium*. Citeseer, 2005, pp. 1–12, doi: https://doi.org/10.31274/etd-180810-1053.

[24] T. Wang, L. Fu, Z. Wei, Y. Zhou, and S. Gao, "Unmanned ground weapon target assignment based on deep q-learning network with an improved multi-objective artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105612, 2023, doi: https://doi.org/10.1016/j.engappai.2022.105612.

[25] H. Na, J. Ahn, and I.-C. Moon, "Weapon–target assignment by reinforcement learning with pointer network," *Journal of Aerospace Information Systems*, vol. 20, no. 1, pp. 53–59, 2023, doi: https://doi.org/10.2514/1.i011150.

[26] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, 2005, pp. 729–734 vol. 2.

[27] S.-Y. Wang and S.-K. Jeng, "A deterministic method for generating a scattering-center model to reconstruct the rcs pattern of complex radar targets," *IEEE Transactions on Electromagnetic Compatibility*, vol. 39, no. 4, pp. 315–323, 1997.

[28] R. E. Ball, *The fundamentals of aircraft combat survivability: analysis and design*. American Institute of Aeronautics and Astronautics, 2003, doi: https://doi.org/10.2514/4.861239.

[29] T. G. Mahnken, *The cruise missile challenge*. Center for Budgetary Assessments, 2005, doi: https://doi.org/10.7249/rr743.

[30] G. W. Byeon, S.-h. Oh, M. J. Kwak, J. Y. Yoon, W. Han, and J. H. Woo, "Development of a methodology for evaluating the susceptibility of naval surface ships based on naval engagement simulation," *Available at SSRN 5097672*, doi: https://doi.org/10.2139/ssrn.5097672.

[31] Z. Wang, Y. Zhang, and B. Yang, "Study on a closed-loop fire correction algorithm in vehicle close-in weapon system," in *2010 International Conference on Computational Intelligence and Software Engineering*. IEEE, 2010, pp. 1–4.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018, doi: https://doi.org/10.1021/acs.jcim.3c01698.s001.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017, doi: https://doi.org/10.1137/1.9781611974409.ch5.

[34] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015, doi: https://doi.org/10.1920/wp.cem.2012.1712.

[35] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1096–1105, doi: https://doi.org/10.1609/aaai.v32i1.11791. [Online]. Available: https://proceedings.mlr.press/v80/dabney18a.html

[36] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2016, doi: https://doi.org/10.21203/rs.3.rs-3440928/v1. [Online]. Available: https://arxiv.org/abs/1511.05952

[37] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1995–2003, doi: https://doi.org/10.1109/access.2024.3380454. [Online]. Available: https://proceedings.mlr.press/v48/wangf16.html

[38] J. Jiang, C. Dun, T. Huang, and Z. Lu, "Graph convolutional reinforcement learning," 2020, doi: https://doi.org/10.5220/0009095207660772.

[39] Y. Zhang, R. Bai, R. Qu, C. Tu, and J. Jin, "A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties," *European Journal of Operational Research*, vol. 300, no. 2, pp. 418–427, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721008821

[40] S. Sun, D. Cai, H.-T. Zhang, and N. Xing, "Reinforcement learning-based mas interception in antagonistic environments," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 1, pp. 270–272, 2024.