2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Transformer–based Hierarchical Reinforcement Learning for Sequential Decision–Making in Swarm Confrontation

Ruozhai Sun[1], Qizhen Wu[2], and Lei Chen[3]

Abstract— **Hierarchical Reinforcement Learning (HRL) is a potent paradigm for addressing long–horizon sequential decision–making in swarm confrontation. However, its strategic capabilities are often bottlenecked by high–level policies that struggle to reason over the dynamic, variable–sized observations of other agents. To address this, we introduce a novel decentralized HRL framework featuring a Transformer–based strategic policy. The Transformer's self–attention mechanism is uniquely suited to capture complex spatio–temporal relationships among a varying number of entities, enabling robust long–horizon task allocation. This high–level strategy is then translated by a low–level policy into collision–free navigation. In complex swarm confrontation scenarios, our method significantly outperforms established baselines, achieving win rates of up to 81%. Beyond this performance, the learned policies exhibit strong zero–shot generalization to larger swarms, offer decision–making interpretability via the attention mechanism, and foster the autonomous emergence of complex cooperative tactics. This work provides a blueprint for scalable, strategically sophisticated, and interpretable multi–agent systems.**

I. INTRODUCTION

Autonomous agent swarms are increasingly pivotal in applications from national defense to search and rescue [1]–[3]. Within this context, swarm confrontation presents a canonical paradigm for advancing decision–making algorithms in complex, mixed cooperative–competitive environments [4]–[6]. Achieving effective autonomous control in these scenarios constitutes an NP–hard problem [4], with its complexity stemming from a confluence of challenges: agents operate with partial observability and constrained communication [7]–[9]; the joint state–action space grows exponentially with the number of agents, inducing a curse of dimensionality; and the credit assignment problem complicates learning from sparse, team-wide rewards [10].

Early strategies for swarm decision–making relied on heuristic and rule–based methods, such as potential fields and expert systems [11]. Although computationally efficient and interpretable, these methods' dependence on hand–crafted rules makes them brittle and unable to adapt to diverse or unforeseen adversarial tactics [12]. To overcome this fragility, the research community has shifted towards learning–based solutions, particularly multi–agent reinforcement learning (MARL) [12]–[14]. Nevertheless, conventional MARL frameworks, such as the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [10], often utilize a flat architecture that maps low–level observations directly to primitive actions. This end–to–end approach struggles in long–horizon scenarios for two key reasons: the difficulty of credit assignment from sparse team–wide rewards [15] and the absence of high–level strategic guidance, which together lead to poor sample efficiency and convergence to suboptimal policies [16]. These shortcomings reveal a fundamental limitation of flat architectures: their inherent lack of temporal abstraction, a capability crucial for effective long–term planning.

HRL directly addresses the need for temporal abstraction by decomposing complex tasks into a hierarchy of sub–policies [4], [17], [18]. This structure is well–suited for swarm confrontation as it facilitates long–horizon planning and simplifies credit assignment [19], [20]. However, a critical bottleneck emerges in multi–agent HRL: empowering the high–level policy to reason over dynamic, variable–sized observations [21]. For instance, an agent's local view may contain a fluctuating number of allies and adversaries. Conventional neural architectures, which require fixed–size inputs, are structurally incompatible with such permutation–invariant and relational entity sets, thereby crippling the strategic capabilities of the entire hierarchy.

To address this challenge, we propose a novel, fully decentralized, two–level hierarchical framework that decomposes the complex agent decision–making problem into high–level strategy and low–level motion control. The core innovation resides in the strategic layer, which features a Transformer–based policy. Its self–attention mechanism is inherently well-suited to reason over the relational structure of variable–sized entity sets, enabling the policy to issue abstract commands such as *Pursuit* or *Escape*. These commands are then translated into concrete spatial waypoints by an intermediate goal selection module using a Task–Aware Potential Field (TAPF), to which a low–level controller executes collision–free navigation. This architecture achieves effective spatio–temporal abstraction, allowing the Transformer–based Q–network to robustly model entity relationships for long–horizon planning. Critically, the attention–based model also provides a powerful mechanism for interpreting the agent's decision–making process. Empirical results demonstrate that our framework not only surpasses established baselines in complex swarm confrontations but also exhibits several key attributes of advanced multi–agent systems: strong zero–shot generalization to larger, unseen scenarios without retraining; enhanced policy interpretability; and the autonomous emer-

This work was supported in part by the National Natural Science Foundation of China under Grants 62088101 and 62003015.

[1]Ruozhai Sun is with the School of Automation, Beijing Institute of Technology, Beijing 100081, China. srz@bit.edu.cn

[2]Qizhen Wu is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China. wuqzh7@buaa.edu.cn

[3]Lei Chen is with the School of Artificial Intelligence, Beijing Institute of Technology, Beijing 100081, China. bit_chen@bit.edu.cn

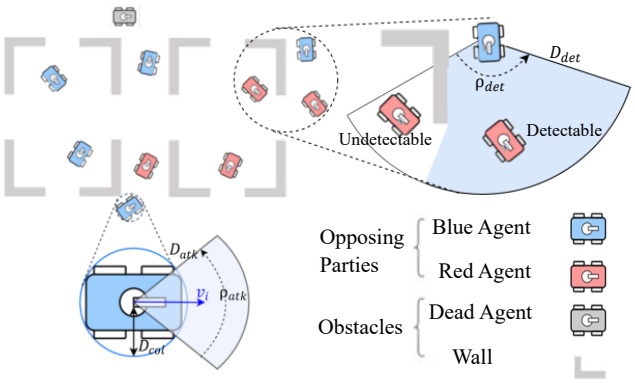


Fig. 1. Illustration of the swarm confrontation scenario and key modeling parameters.

gence of sophisticated cooperative strategies.

The remainder of this paper is organized as follows. Section II formulates the swarm confrontation problem. Section III details the architecture and training of our proposed hierarchical framework. Section IV presents extensive simulation results and generalization studies. Finally, Section V concludes the paper.

II. PROBLEM FORMULATION

This research investigates swarm confrontation through a simulated urban combat scenario. The simulation features two opposing, homogeneous teams competing within a bounded, obstacle-populated 2D environment. As depicted in Fig. 1, the primary objective for each team is to neutralize opposing agents through coordinated engagement while simultaneously avoiding collisions with teammates and obstacles. Each episode commences with agents being randomly initialized in designated map regions, which guarantees a minimum initial separation distance of D_{initial} between any two entities.

Each agent is modeled with second-order kinematics, its state defined by position $\boldsymbol{p} \in \mathbb{R}^2$, velocity $\boldsymbol{v} \in \mathbb{R}^2$, and acceleration $\boldsymbol{a} \in \mathbb{R}^2$. Motion is subject to a maximum velocity $v_{\max}$ and acceleration $a_{\max}$ along each axis. To emulate realistic operational limits, each agent can communicate with up to n_c neighbors within a communication range D_c. Through this channel, agents share their state information. Each agent perceives its surroundings via a simulated LiDAR with a detection range of D_{det} and a field of view of ρ_{det}, and can engage targets only within an attack range of D_{atk} and angle of ρ_{atk}. All agents have a physical collision radius of D_{col}; neutralized agents become static obstacles. The scenario unfolds over a fixed time horizon of $t_{\max}$, with the winner being the team that maximizes enemy eliminations while minimizing its own losses.

To formally address this complex cooperative task, we model it as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. Here, $\mathcal{N}$ is the set of n cooperative agents. At each timestep t, the environment is in a global state $s_t \in \mathcal{S}$ not fully accessible to the agents. Instead, each agent $i \in \mathcal{N}$ receives a local observation $o_{i,t} \in \mathcal{O}$. This observation is composed of the agent's direct sensor readings and the information received from its teammates. Based on this, the agent selects an action $a_{i,t} \in \mathcal{A}_i$. The resulting joint action $\boldsymbol{a}_t \in \mathcal{A}$ induces a state transition according to the function $\mathcal{P}(s_{t+1}|s_t, \boldsymbol{a}_t)$, and the team receives a shared reward via the function $\mathcal{R}(s_t, \boldsymbol{a}_t)$.

The team's collective objective is to learn a joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ that maximizes the expected discounted return, formulated as

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0}^{T_{\max}} \gamma^t \mathcal{R}(s_t, \boldsymbol{a}_t) \right], \tag{1}$$

where $J(\boldsymbol{\pi})$ is the objective function, $t_{\max}$ is the episode horizon, $\gamma \in [0, 1)$ is the discount factor, and $\mathcal{R}(s_t, \boldsymbol{a}_t)$ is the team-wide reward at timestep t.

III. METHODOLOGY

To address the complex decision-making challenges in swarm confrontation, this paper introduces a novel, decentralized hierarchical reinforcement learning framework. Our approach decomposes an agent's control problem into a cascade of interconnected modules. At the highest level, a Transformer-based strategic policy performs long-horizon reasoning to select a macroscopic task. This directive is then translated into a specific spatial waypoint by an intermediate goal selection model. Finally, a low-level motion policy generates continuous velocity commands to navigate towards this waypoint while avoiding collisions. This hierarchical structure effectively integrates long-term strategic planning with immediate reactive control, providing a robust solution for coordinated multi-agent behavior. The overall architecture is illustrated in Fig. 2.

A. High–Level Strategic Policy

The high–level policy π_θ operates at a lower temporal frequency to determine an agent's strategic intent. It approximates the optimal action–value function $Q^*(o_{i,t}, a^H_{i,t})$ using a Transformer–based Q–network. This network processes an agent's local observation sequence $o_{i,t}$ to select a discrete strategic task $a^H_{i,t}$ from the predefined set $\mathcal{Z} = \{\textit{Search, Pursuit, Escape, Support}\}$.

1) Network Architecture: As depicted in Fig. 2, the Transformer Q–network is designed to handle the variable and unordered nature of multi–agent observations. It processes a set of feature vectors $\boldsymbol{E}_{i,t} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{K_t}\}$, corresponding to the K_t entities an agent perceives at timestep t. Each entity's feature vector $\boldsymbol{x}_j$ is first projected into a high–dimensional embedding and then augmented with a sinusoidal positional encoding $\boldsymbol{p}_j$ to retain spatial information:

$$\boldsymbol{x}'_j = \text{Embedding}(\boldsymbol{x}_j) + \boldsymbol{p}_j. \tag{2}$$

A multi-layer Transformer encoder then processes the sequence of these augmented vectors. A key challenge is handling the variable number of entities an agent perceives at any given time. Our architecture addresses this by applying a dynamic attention mask matrix $\boldsymbol{M}$ during self-attention.

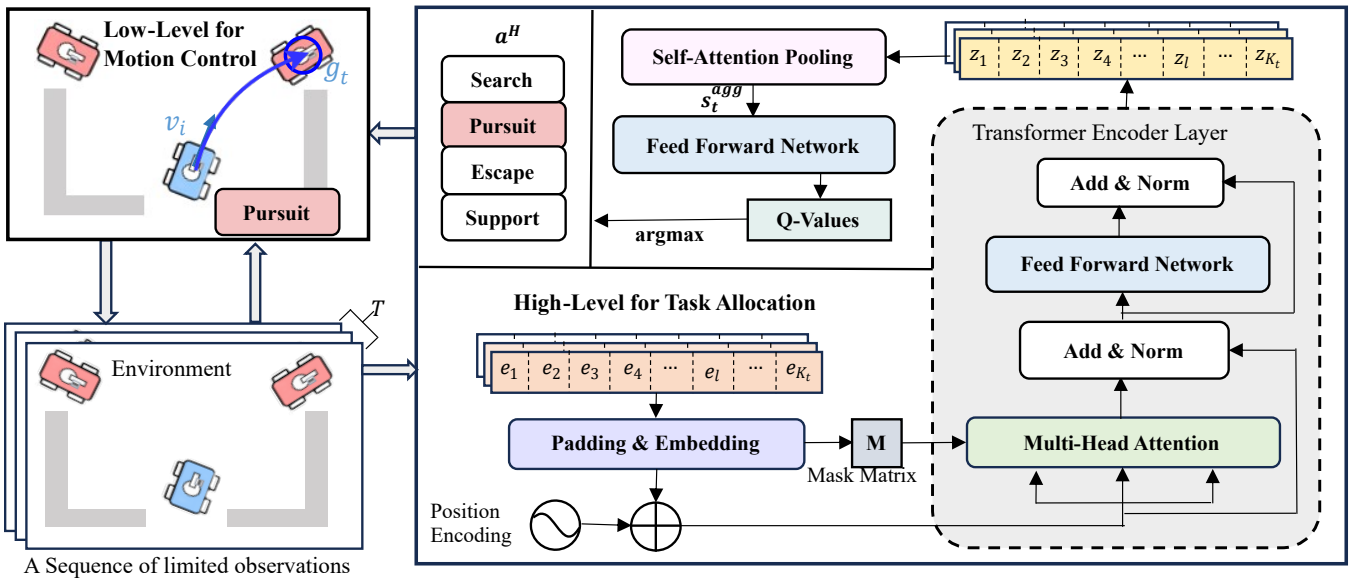


Fig. 2. This figure illustrates the overall architecture of the proposed HRL framework, where the high–level, Transformer–based policy selects a strategic task (a^H), which is translated into a waypoint ($\boldsymbol{g}_t$) and then executed by the low–level motion controller via continuous velocity commands ($\boldsymbol{v}_{i,t}$).

This mask nullifies the influence of any padded tokens (used to create uniform tensor sizes), thereby ensuring the model's focus remains strictly on valid entity information.

Using scaled dot-product attention, we implement the self-attention mechanism, where d_k is the dimension of the key vectors. Its output is calculated as

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}} + \boldsymbol{M}\right)\boldsymbol{V}. \quad (3)$$

The encoder's output $\boldsymbol{Z}_t$, which contains contextually–aware representations of all entities, is aggregated by a self–attention pooling mechanism into a single fixed–dimension vector $\boldsymbol{s}_t^{\text{agg}}$. This vector serves as a tactical summary of the local situation and is passed to a multi–layer perceptron (MLP), which functions as the feed–forward network (FFN), to compute the final Q–values for each possible task:

$$Q(o_{i,t}, a^H; \theta) = \text{MLP}(\boldsymbol{s}_t^{\text{agg}}). \quad (4)$$

2) Reward Formulation: The strategic policy is trained via a hybrid reward R_H balancing team-wide success r_{global} and dense heuristic shaping r_d:

$$R_H = \lambda_g r_{\text{global}} + \lambda_d r_d, \quad (5)$$

where λ_g and λ_d are weighting coefficients. The global reward r_{global} provides a direct measure of mission success and is defined as the change in the team's total enemy elimination count over a high-level decision period.

For immediate, localized feedback to guide policy exploration, we introduce a shaping reward $r_d \in \{-1, 0, 1\}$. Its value is derived from the sign of a heuristic advantage function, $adv(o_{i,t})$, which estimates the tactical soundness of an agent's current situation. This function aggregates the pairwise advantage $p_{j,i}$ that agent i holds over every enemy j within its set of perceived entities $\mathcal{E}_{i,t}$:

$$adv(o_{i,t}) = \sum_{j \in \mathcal{E}_{i,t}} p_{j,i}. \quad (6)$$

Central to this heuristic is the pairwise advantage metric $p_{j,i}$, formulated to capture both geometric superiority and proximity priority:

$$p_{j,i} = \frac{\cos\theta_{i,j} - \cos\theta_{j,i}}{2 + d_{i,j}}, \quad (7)$$

where $d_{i,j}$ represents the Euclidean distance between agent i and enemy j, and $\cos\theta_{i,j}$ denotes the cosine similarity between agent i's velocity vector $\boldsymbol{v}_i$ and the relative position vector $\boldsymbol{d}_{i,j}$ (from i to j).

This metric's design is rooted in two tactical principles. The numerator $\cos\theta_{i,j} - \cos\theta_{j,i}$ quantifies angular superiority, maximizing its value when agent i achieves a favorable engagement posture. Concurrently, the denominator provides proximity weighting, ensuring that more immediate threats and opportunities are prioritized.

3) Policy Training: Training of the Q-network, parameterized by θ, is achieved by minimizing the Mean-Squared Bellman Error (MSBE) loss $\mathcal{L}(\theta)$:

$$\mathcal{L}(\theta) = \mathbb{E}_{(o_{i,t}, a_{i,t}^H, R_H, o_{i,t+1}) \sim \mathcal{D}_H}\left[\left(y_t - Q(o_{i,t}, a_{i,t}^H; \theta)\right)^2\right], \quad (8)$$

where the target value y_t is computed using a target network Q' (with parameters θ') to enhance training stability:

$$y_t = R_H + \gamma \max_{a' \in \mathcal{Z}} Q'(o_{i,t+1}, a'; \theta'). \quad (9)$$

Following the loss computation, the policy network's parameters are updated via stochastic gradient descent, while the target network's parameters are updated using a soft replacement rule:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta), \quad (10)$$

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta'. \tag{11}$$

B. Low-Level Motion Policy

Operating at the highest temporal frequency, the low-level motion policy μ_ψ translates a spatial waypoint $\boldsymbol{g}_t$ into a continuous velocity command $\boldsymbol{v}_{i,t}$ for smooth and safe navigation. We implement this policy using an actor-critic framework based on the MADDPG algorithm, enhanced with Prioritized Experience Replay (PER) for improved sample efficiency. During execution, each agent's actor policy operates in a fully decentralized manner, conditioned only on its local observation $o_{i,t}$ and its assigned waypoint $\boldsymbol{g}_t$.

The learning process is driven by a composite intrinsic reward function $R_{L,i}$, which is independent of the high-level task reward R_H and is designed to provide dense and sparse feedback for efficient subgoal completion. This functional decoupling allows the low-level policy to be pre-trained as a general-purpose navigation controller. The total intrinsic reward for agent i at timestep t is a weighted sum of three distinct components:

$$R_{L,i,t} = w_p r_{\text{prog},i,t} + w_s r_{\text{succ},i,t} - w_c r_{\text{coll},i,t}, \tag{12}$$

where w_p, w_s, and w_c are positive weighting coefficients. These are set empirically to balance the influence of dense progress-shaping rewards with sparse signals for critical events, a configuration that proved robust across all of our tested scenarios.

The three components of the reward function serve distinct purposes. A dense progress-shaping reward r_{prog} encourages continuous movement towards the goal, and is based on the reduction of distance over a single discrete timestep:

$$r_{\text{prog},i,t} = ||\boldsymbol{p}_{i,t-1} - \boldsymbol{g}_t||_2 - ||\boldsymbol{p}_{i,t} - \boldsymbol{g}_t||_2, \tag{13}$$

where $\boldsymbol{p}_{i,t}$ is the agent's position at timestep t, and $||\cdot||_2$ denotes the Euclidean distance. In contrast, two sparse signals provide feedback for critical events. A success bonus r_{succ} is awarded for reaching the goal's vicinity, while a collision penalty r_{coll} is applied for any unsafe movements. These rewards are formulated via an indicator function:

$$r_{\text{succ},i,t} = \mathbb{I}(||\boldsymbol{p}_{i,t} - \boldsymbol{g}_t||_2 < d_{\text{succ}}), \tag{14}$$

$$r_{\text{coll},i,t} = \mathbb{I}(\text{collision}_{i,t}), \tag{15}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns a value of 1 if its condition is true and 0 otherwise. The term d_{succ} is the distance threshold for defining success, and $\text{collision}_{i,t}$ is a boolean flag indicating whether agent i experienced a collision at timestep t.

The centralized critic for each agent i is updated by minimizing a standard Mean-Squared Error (MSE) loss:

$$\mathcal{L}(\omega_i) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{a},r,\boldsymbol{x}')\sim\mathcal{D}_L}\left[\left(y_i - Q_{\omega_i}(\boldsymbol{x},\boldsymbol{a})\right)^2\right], \tag{16}$$

where $y_i = r_{L,i,t} + \gamma Q'_{\omega'_i}(\boldsymbol{x}', \boldsymbol{a}'|_{\boldsymbol{a}'_j=\mu'_{\psi'_j}(o_j)})$ is the target value calculated using the target networks. The corresponding actor is updated using the deterministic policy gradient:

$$\nabla_{\psi_i} J(\psi_i) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_L}\left[\nabla_{\psi_i}\mu_{\psi_i}(o_i)\nabla_{a_i}Q_{\omega_i}(\boldsymbol{x},\boldsymbol{a})|_{a_i=\mu_{\psi_i}(o_i)}\right]. \tag{17}$$

Finally, the actor's policy parameters are updated using the deterministic policy gradient, which provides a robust and adaptive foundation for motion execution.

C. Embedded Goal Selection Model

This intermediate model is responsible for generating a spatial waypoint $\boldsymbol{g}_t$ based on the high-level task command $a^H_{i,t}$. It achieves this by constructing a Task-Aware Potential Field (TAPF), where the field's construction logic is tailored to the specific task assigned.

1) Goal Selection for Pursuit and Support Tasks: For the *Pursuit* and *Support* tasks, where the objective is to close the distance to a specific agent, the goal $\boldsymbol{g}_t$ is simply set to the target's last known position $\boldsymbol{p}_{\text{target},t}$:

$$\boldsymbol{g}_t = \boldsymbol{p}_{\text{target},t}. \tag{18}$$

2) Goal Selection for Search Task: When assigned the *Search* task, an agent must explore the environment efficiently. A potential field $U_{\text{search}}(\boldsymbol{p})$ is constructed by combining an attractive potential $U_{\text{att,s}}$ (promoting dispersed exploration away from the team's center of mass) and a repulsive potential $U_{\text{rep,ent}}$ (preventing collisions):

$$U_{\text{att,s}}(\boldsymbol{p}) = 0.5k_{\text{att}}||\boldsymbol{p} - \boldsymbol{p}_{\text{team}}||^2, \tag{19}$$

$$U_{\text{rep,ent}}(\boldsymbol{p}) = \begin{cases} 0.5k_{\text{rep}}\left(\frac{1}{d(\boldsymbol{p},\boldsymbol{p}_{\text{ent}})} - \frac{1}{d_0}\right)^2 & \text{if } d(\boldsymbol{p},\boldsymbol{p}_{\text{ent}}) \le d_0, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

The final waypoint $\boldsymbol{g}_t$ is set a fixed distance D_{goal} from the agent's current position $\boldsymbol{p}_t$ in the direction of the net force, $\boldsymbol{F}_{\text{search}} = -\nabla(U_{\text{att,s}} + \sum U_{\text{rep,ent}})$:

$$\boldsymbol{g}_t = \boldsymbol{p}_t + D_{\text{goal}}\frac{\boldsymbol{F}_{\text{search}}}{||\boldsymbol{F}_{\text{search}}||}. \tag{21}$$

3) Goal Selection for Escape Task: For the *Escape* task, the potential field $U_{\text{escape}}(\boldsymbol{p})$ is purely repulsive, constructed by summing the potentials $U_{\text{rep,ent}}(\boldsymbol{p})$ from all perceived threats. The waypoint $\boldsymbol{g}_t$ is then set a fixed distance D_{escape} in the direction of the net repulsive force $\boldsymbol{F}_{\text{escape}}$:

$$\boldsymbol{g}_t = \boldsymbol{p}_t + D_{\text{escape}}\frac{\boldsymbol{F}_{\text{escape}}}{||\boldsymbol{F}_{\text{escape}}||}. \tag{22}$$

The potential field parameters ($k_{\text{att}}, k_{\text{rep}}, d_0, D_{\text{goal}}, D_{\text{escape}}$) are determined empirically and a single, fixed set of values is used across all scenarios, demonstrating the robustness of this model without requiring environment-specific tuning.

D. Two-Stage Training Procedure

To ensure stability and sample efficiency, we employ a two-stage training procedure as detailed in Algorithm 1. This staged approach decouples the complex learning problem into two more manageable sub-problems, leading to faster convergence and more robust final policies.

The initial stage is dedicated to pre–training the low-level policy. In this phase, the MADDPG–based motion controller is trained independently to navigate toward randomly generated waypoints. This allows it to master the fundamental skill

Algorithm 1 Hierarchical Training Procedure

Input: Epochs $E_{\text{pre}}, E_{\text{main}}$.
Initialization: Policy parameters $\theta, \{\psi_i\}, \{\omega_i\}$; Replay buffers $\mathcal{D}_H, \mathcal{D}_L$.

▷ Stage 1: Pre-train Low-Level Policy
for $e = 1 \rightarrow E_{\text{pre}}$ **do**
  Collect low-level experiences into $\mathcal{D}_L$ by executing μ_ψ with random goals.
  Sample a batch of transitions from $\mathcal{D}_L$.
  $\mathcal{L}(\omega_i) \leftarrow$ Eq. 16, $\nabla_{\psi_i} J(\psi_i) \leftarrow$ Eq. 17.
  Refresh networks $Q_{\omega_i}, \mu_{\psi_i}$.
end for
Freeze parameters $\{\psi_i, \omega_i\}$.
▷Stage 2: Train High-Level Policy
for $e = 1 \rightarrow E_{\text{main}}$ **do**
  Collect macro-transitions into $\mathcal{D}_H$ by executing π_θ.
  Sample a batch of macro-transitions from $\mathcal{D}_H$.
  $\mathcal{L}(\theta) \leftarrow$ Eq. 8.
  Refresh network Q_θ via $\nabla_\theta \mathcal{L}(\theta)$.
end for

of robust, collision-free movement without the complexities of strategic decision-making. Once the low–level policy is proficient, its parameters are frozen, and the training proceeds to the second stage. Here, we train the high–level Transformer Q-network end-to-end. During this stage, the pre–trained motion controller effectively becomes part of the environment dynamics from the perspective of the strategic policy. This decoupling is critical, as it allows the high–level policy to focus exclusively on learning long–horizon strategy without being burdened by low–level control.

IV. EXPERIMENTS

This section presents a rigorous, multi–faceted validation of our proposed HRL framework. We first establish its superior combat effectiveness by benchmarking against established baselines, followed by targeted ablation studies to verify the critical contributions of its core architectural components. We then assess the framework's practical scalability through demanding zero–shot generalization tests on larger, unseen swarm scenarios. The evaluation culminates in a qualitative analysis that leverages the Transformer's attention mechanism to reveal two hallmarks of advanced multi–agent intelligence: the interpretability of individual agent decisions and the emergence of sophisticated cooperative strategies at the swarm level.

A. Experimental Setup

1) Environment and Scenarios: All simulations are conducted in a 30×16 rectangular area populated with a fixed configuration of static obstacles, designed to simulate an urban environment. To evaluate scalability, we test three distinct confrontation scales: 5–versus–5 (V5), 9–versus–9 (V9), and 13–versus–13 (V13). In each scenario, our trained agents (the blue team) compete against an enemy red team controlled by a deterministic, rule–based policy. This ensures a consistent and repeatable challenge for fair comparison. Each agent starts with 5 health points, and an episode terminates after $t_{\max} = 200$ timesteps or when all agents on one team are eliminated. Key agent parameters, which are consistent across all scales, are detailed in Table I.

TABLE I
KEY AGENT PARAMETERS IN THE SIMULATION.

Category	Parameter	Value
Kinematics	Maximum Velocity ($v_{\max}$)	1.0 m/s
	Maximum Acceleration ($a_{\max}$)	1.0 m/s^2
Perception	Enemy Detection Range (D_{det})	5.0 m
	Enemy Detection Angle (ρ_{det})	150°
Interaction	Ally Communication Range (D_{com})	10.0 m
	Attack Range (D_{atk})	1.0 m
	Collision Radius (D_{col})	0.3 m
State	Initial Health Points (HP)	5

2) Implementation Details: The high–level policy consists of a 2–layer Transformer with 4 attention heads and a model dimension of 64. The low–level actor and critic networks are MLPs with two hidden layers of 128 units each. Both policies use the Adam optimizer with a discount factor $\gamma = 0.99$ and a target network update rate $\tau = 0.01$. The high–level policy uses a learning rate of 1×10^{-4} and a batch size of 512, while the low–level policy uses a learning rate of 1×10^{-3} and a batch size of 1024.

B. Comparative Analysis

To provide a comprehensive performance evaluation, our HRL-T framework is benchmarked against a diverse set of competitors. These competitors are carefully selected to represent a spectrum of distinct algorithmic paradigms, allowing for a multi-faceted assessment of our method's capabilities. They include a strong, non-learning Expert System; a conventional, non-hierarchical Deep Reinforcement Learning (DRL) agent with a flat architecture; and HRL-D, a direct hierarchical counterpart that employs a standard Deep Q-Network (DQN) instead of a Transformer. This final comparison is particularly crucial for isolating the specific advantages conferred by our Transformer-based high-level policy. The competitors are introduced as follows:

- Expert System: A strong non-learning competitor using handcrafted, hierarchical rules based on established tactical principles [22].
- DRL: A non-hierarchical, end-to-end approach based on MADDPG [10], which learns a flat policy mapping joint observations directly to continuous actions.
- HRL–D: An ablation of our framework where the high–level Transformer is replaced by a DQN to evaluate the Transformer's contribution.

For a fair and consistent evaluation, all methods were tested under an identical experimental setup, using the same input data and performance metrics. An analysis of the learning dynamics, presented in Fig. 3, reveals the superior learning efficiency of our HRL–T method. Across all tested swarm scales, HRL–T consistently demonstrates the fastest

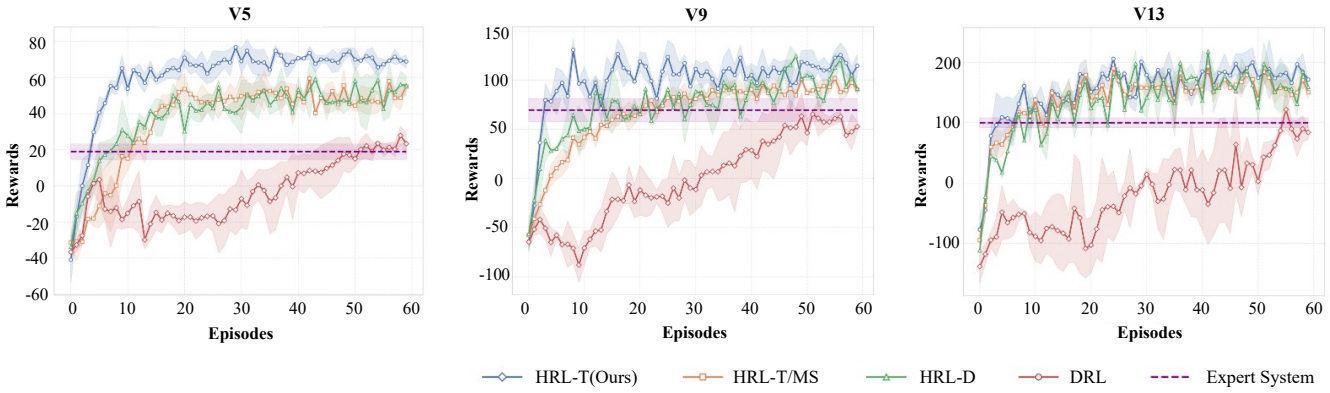


Fig. 3. Learning curves comparing our method (HRL–T) with baselines (DRL, HRL–D, Expert System) and an ablated version (HRL–T/MA) across different swarm sizes (V5, V9, V13). Solid curves represent the mean rewards over all training instances, while the shaded regions indicate the standard deviation.

TABLE II
EXPERIMENTAL RESULTS OF BASELINES AND OUR METHOD FOR DIFFERENT SWARM SIZES

Method	V5			V9			V13		
	Re.	W.R.(%)	K/D	Re.	W.R.(%)	K/D	Re.	W.R.(%)	K/D
Expert System	21.0 ± 5.2	52 ± 4	1.05 ± 0.06	70.0 ± 15.8	50 ± 2	1.00 ± 0.10	100.0 ± 10.1	50 ± 2	1.02 ± 0.08
DRL	22.5 ±10.7	55 ± 6	1.10 ± 0.09	55.2 ± 30.1	47 ± 7	0.92 ± 0.20	85.2 ± 18.2	41 ± 3	0.80 ± 0.14
HRL–D	55.3 ± 15.1	75 ± 4	1.32 ± 0.10	115.8 ± 25.6	**77 ± 3**	1.36 ± 0.17	165.3 ± 25.6	70 ± 1	1.28 ± 0.17
HRL–T/MA	55.5 ± 8.7	75 ± 2	1.32 ± 0.07	108.2 ± 18.8	74 ± 5	1.31 ± 0.14	152.6 ± 20.3	67 ± 3	1.20 ± 0.15
HRL–T/RS	–	40 ± 4	0.82 ± 0.10	–	44 ± 2	0.90 ± 0.15	–	52 ± 3	1.02 ± 0.10
HRL–T	**73.1 ± 10.2**	**81 ± 2**	**1.48 ± 0.07**	**120.6 ± 20.4**	76 ± 1	**1.37 ± 0.14**	**180.0 ± 25.1**	**71 ± 2**	**1.28 ± 0.17**

convergence and sustains the highest reward plateau. In contrast, the non–hierarchical DRL baseline struggles to scale, with its performance deteriorating as the number of agents increases. While the HRL–D variant learns effectively, it does not achieve the same performance level as the full Transformer–based framework.

The quantitative results in Table II further confirm HRL–T's strategic superiority, showing that it consistently secures the highest average rewards and Kill/Death (K/D) ratios. This dominance is particularly evident in the complex V13 scenario, where HRL–T achieves a leading K/D ratio of 1.28 while the DRL baseline's performance degrades significantly to 0.80. This robust performance across key strategic metrics affirms the advanced capabilities of our framework.

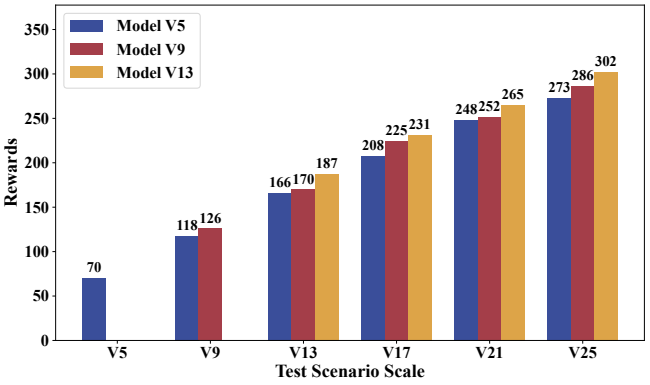


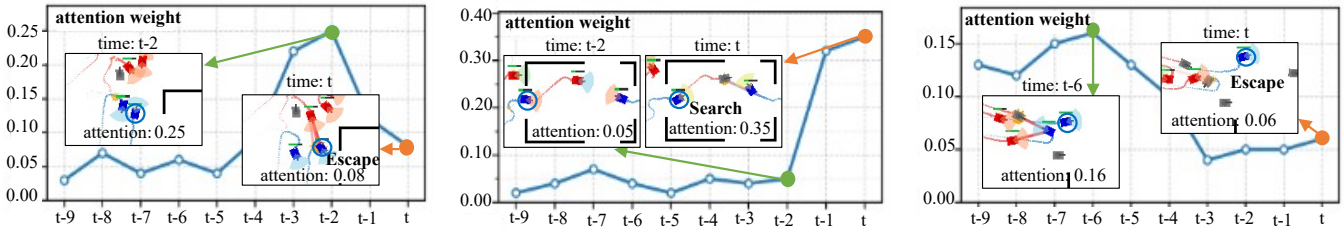
Fig. 4. Generalization performance of models trained on V5, V9, and V13 scenarios and tested on larger, unseen scales.

C. Ablation Studies

We conducted a series of ablation studies to rigorously validate the contributions of the core components within our proposed HRL–T framework. These studies involved evaluating the following two modified versions of our model:

- HRL–T/MA: The HRL–T model without the dynamic mask attention mechanism in the high-level policy. In this variant, the attention mechanism processes padded inputs without masking.
- HRL–T/RS: The HRL–T model trained without reward shaping. This version relies solely on the sparse global reward for learning the high–level task allocation policy.

The performance of both ablated models is detailed in Fig. 3 and Table II, providing a clear basis for analysis. An examination of the HRL–T/MA variant, which lacks the dynamic mask, reveals a distinct performance deficit. The learning curves in Fig. 3 show that this variant exhibits slower convergence and reaches a lower reward plateau. This finding is corroborated by the quantitative data in Table II, which indicates that its final K/D ratio and win rates are consistently inferior to the full HRL–T model across all scenarios. This performance gap unequivocally underscores the importance of the masking mechanism; by enabling the Transformer to focus its attention on relevant tactical



(a) Act on remembered threats. (b) Adapt to contextual shifts. (c) Leverage teammate intelligence.

Fig. 5. Visualization of the attention mechanism linking weights to in–game events across three illustrative scenarios.The line graphs show the attention weights an agent places on its last 10 observations (from $t-9$ to t) when making a decision at the current timestep t.

information, the mask proves instrumental in accelerating convergence and fostering a more robust final policy.

The necessity of our hybrid reward structure becomes even more apparent when analyzing the HRL–T/RS variant. Trained using only the sparse global reward, this model struggles to learn a viable policy, a failure confirmed by its markedly poor performance in Table II, with K/D ratios near or below 1.0 and low win rates. This outcome highlights the severe challenge of credit assignment when learning from a solely delayed, team-based signal. In contrast, the dense, heuristic–based decision-shaping reward (r_d) is crucial, as it provides the immediate feedback needed to effectively guide policy exploration and prevent agents from becoming trapped in suboptimal strategies. Collectively, these ablation studies empirically validate that both the dynamic attention mask and the hybrid reward shaping are integral and indispensable components for the success of the HRL–T framework.

D. Generalization to Larger–Size Swarms

In practical applications, retraining models for every potential swarm size is computationally prohibitive. Therefore, it is crucial to evaluate the ability of policies trained on smaller scales to generalize to larger, unseen scenarios. To this end, we conducted a generalization experiment where we took the fully trained HRL–T policies from the V5, V9, and V13 scenarios and deployed them directly, without any fine–tuning, in a series of progressively larger environments: V17, V21, and V25. The performance of each model was measured by the total rewards.

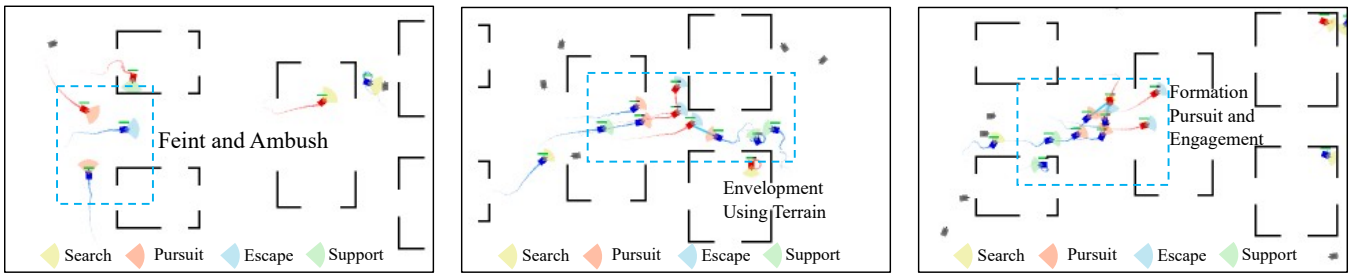
Fig. 4 highlights the framework's remarkable zero–shot generalization. Our policies, trained on smaller scales, exhibit robust performance when transferred to larger, unseen scenarios without any fine–tuning. For example, the policy trained on the V5 scenario, when deployed in the V13 environment, achieves a higher reward than baseline methods that were specifically trained for V13. This result strongly validates that our architecture learns inherently scalable strategies. While models trained on more complex scenarios expectedly perform best in the largest tests, the impressive scalability of even the simplest model underscores the framework's practical utility, alleviating the need for retraining for every potential swarm size.

E. Analysis of Learned Policies: Interpretability and Emergent Strategies

A primary challenge in deep reinforcement learning is the black box nature of learned policies. Our framework mitigates this by leveraging the Transformer's self–attention mechanism as a powerful tool for interpretability. By quantitatively analyzing the attention weights an agent places on its recent observational history (a sequence of 10 timesteps), we can gain insight into its decision–making rationale. The analysis, illustrated in Fig. 5, reveals that the policy learns logical and context–aware attention patterns. For instance, agents learn to act on remembered threats (Fig. 5a), with an *Escape* decision being primarily influenced by observations from several timesteps prior when an enemy was last visible. The policy also adapts to contextual shifts (Fig. 5b), such as when an agent, after neutralizing a threat, switches to a *Search* task and its attention immediately focuses on the most recent, non–hostile observations. Furthermore, agents can leverage teammate intelligence (Fig. 5c), initiating an escape by attending to older information that corresponds to a threat relayed by an ally.

This capacity for selective temporal integration confirms that the attention mechanism endows agents with a sophisticated, context–aware reasoning ability. It dynamically prioritizes crucial information while filtering out irrelevant data. This not only validates the use of the Transformer architecture for high–level policy learning but also demonstrates its potential for fostering robust and, critically, interpretable decision–making in complex multi–agent scenarios.

Beyond individual agent interpretability, our framework also fosters the emergence of sophisticated cooperative strategies at the swarm level. These complex, coordinated tactics are not pre–programmed but are learned autonomously through end–to–end training, demonstrating true emergent intelligence. As illustrated in Fig. 6, the trained swarm exhibits a rich repertoire of tactical behaviors. These include executing a feint–and–ambush maneuver (Fig. 6a), leveraging terrain for tactical envelopment (Fig. 6b), and maintaining a cohesive formation to pursue and concentrate fire on adversaries (Fig. 6c). The autonomous development of such strategies underscores the framework's ability to derive effective, high–level cooperative solutions.



(a) V5 (b) V9 (c) V13

Fig. 6. Visualization of emergent cooperative strategies across different scales.

V. CONCLUSION

We have presented a two–level HRL framework to tackle the complex challenge of swarm confrontation. Our core contribution, a Transformer–based strategic policy, has proven effective for reasoning over variable local observations, leading to superior combat effectiveness compared to baseline methods. The framework has also fostered the autonomous emergence of complex cooperative tactics, has shown strong zero–shot generalization to larger scales, and has provided policy interpretability. While this work validates our approach as a significant step toward scalable and strategically capable multi–agent systems, we acknowledge the computational cost of self-attention as a key limitation. Future efforts should therefore be directed toward exploring more efficient relational architectures to address this challenge.

REFERENCES

[1] Z. Du, C. Luo, G. Min, J. Wu, C. Luo, J. Pu, and S. Li, "A survey on autonomous and intelligent swarms of uncrewed aerial vehicles (uavs)," *IEEE Trans. Intell. Transport. Syst.*, pp. 1–24, 2025.

[2] M. Kouzeghar, Y. Song, M. Meghjani, and R. Bouffanais, "Multi-target pursuit by a decentralized heterogeneous UAV swarm using deep multi-agent reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 3289–3295.

[3] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (uavs): practical aspects, applications, open challenges, security issues, and future trends," *Intel Serv Robotics*, vol. 16, no. 1, pp. 109–137, Mar. 2023.

[4] Q. Wu, K. Liu, L. Chen, and J. Lü, "Hierarchical reinforcement learning for swarm confrontation with high uncertainty," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 8630–8644, 2024.

[5] J. Liu, G. Wang, Q. Fu, S. Yue, and S. Wang, "Task assignment in ground-to-air confrontation based on multiagent deep reinforcement learning," *Defence Technology*, vol. 19, pp. 210–219, Jan. 2023.

[6] B. Yang, L. Mo, M. Lv, J. Wang, and C. Chen, "Exploring critical decision schemes and tactical parameters in typical beyond visual range air combat," *IEEE Trans. Veh. Technol.*, vol. 74, no. 1, pp. 348–361, Jan. 2025.

[7] W. Xia, Z. Zhou, W. Jiang, and Y. Zhang, "Dynamic uav swarm confrontation: An imitation based on mobile adaptive networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 5, pp. 7183–7202, 2023.

[8] C. De Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized multi-agent pursuit using deep reinforcement learning," *IEEE Rob. Autom. Lett.*, vol. 6, no. 3, pp. 4552–4559, Jul. 2021.

[9] T. Phan, F. Ritz, P. Altmann, M. Zorn, J. Nüßlein, M. Kölle, T. Gabor, and C. Linnhoff-Popien, "Attention-based recurrence for multi-agent reinforcement learning under stochastic partial observability," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 202, Jul 2023, pp. 27 840–27 853.

[10] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[11] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Front. Robot. AI*, vol. 7, Apr. 2020.

[12] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 895–943, 2022.

[13] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[14] A. Gupta, J. Kubička, and M. Šepek, "Solving large-scale pursuit-evasion games using pre-trained strategies," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 37, no. 10, 2023, pp. 12 151–12 159.

[15] M. Dawood, N. Dengler, J. de Heuvel, and M. Bennewitz, "Handling sparse rewards in reinforcement learning using model predictive control," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 879–885.

[16] T. Zhang, L. Chai, S. Wang, J. Jin, X. Liu, A. Song, and Y. Lan, "Improving autonomous behavior strategy learning in an unmanned swarm system through knowledge enhancement," *IEEE Trans. Rel.*, vol. 71, no. 2, pp. 763–774, June 2022.

[17] S. Pateria, B. Subagdja, A.-H. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–35, 2022.

[18] X. Mao, G. Wu, M. Fan, Z. Cao, and W. Pedrycz, "DL-DRL: A double-level deep reinforcement learning approach for large-scale task scheduling of multi-uav," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 1028–1044, 2024.

[19] B. Li, J. Li, T. Lu, Y. Cai, and S. Wang, "Hierarchical learning from demonstrations for long-horizon tasks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 4545–4551.

[20] T. Ma, K. Peng, H. Rong, Y. Qian, and N. Al-Nabhan, "Hierarchical coordination multi-agent reinforcement learning with spatio-temporal abstraction," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 8, no. 1, pp. 533–547, Feb 2024.

[21] Z. Zhang, D. Zhang, Q. Zhang, W. Pan, and T. Hu, "DACOOP-A: Decentralized adaptive cooperative pursuit via attention," *IEEE Robot. Autom. Lett.*, vol. 9, no. 6, pp. 5504–5511, 2023.

[22] Y. Hou, X. Liang, J. Zhang, M. Lv, and A. Yang, "Hierarchical decision-making framework for multiple ucavs autonomous confrontation," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 13 953–13 968, Nov 2023.