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

Graph-Based Multi-Agent Reinforcement Learning for Scalable UAV Formation Control and Target Tracking

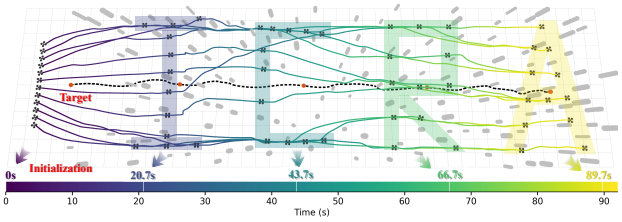
Haowen Wang[1], Shuting Zhang[1], and Guangchen Li[1]

***Abstract*— This paper presents a graph-based multi-agent reinforcement learning framework for scalable UAV formation control and target tracking. The framework introduces a conflict-aware graph representation that aggregates neighborhood information through attention-based message passing, enabling each UAV to analyze both local interactions and global formation geometry. To generate agile and stable maneuvers, a hierarchical policy is designed that first selects motion primitives from a structured library and then refines them with continuous trajectory adjustments, ensuring smooth and dynamically feasible flight in cluttered environments. Extensive simulations and real-world experiments validate the proposed approach, demonstrating accurate target tracking, stable formation maintenance, and robust adaptation across varying swarm sizes and obstacle densities. In particular, policies trained on smaller swarms generalize effectively to larger ones without retraining, highlighting the scalability and practicality. The demonstration video is available on the project website: https://swift520.github.io/Formation-Tracking/.**

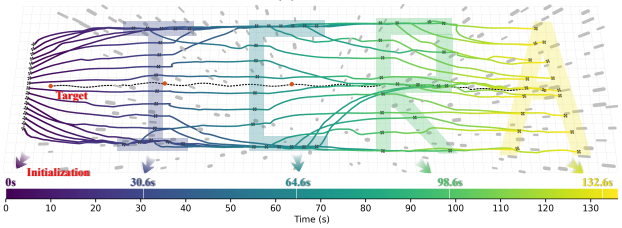
I. Introduction

Unmanned Aerial Vehicle (UAV) swarms are increasingly deployed in missions such as surveillance, environmental monitoring, and search-and-rescue, where multiple agents must coordinate in dynamic environments [1]. A key requirement is the ability to track moving targets while maintaining prescribed formations. Target tracking ensures persistent observation of the object of interest, whereas formation control provides spatial organization that supports sensing coverage, communication efficiency, and collision avoidance. In practice, these two objectives are tightly coupled: tracking deteriorates without stable formations, while rigid formations may fail in the presence of agile targets or obstacles. This motivates unified strategies that jointly address formation control and target tracking, enabling robust and adaptive swarm operation in cluttered environments.

Classical approaches such as Artificial Potential Fields (APF) [2] and Virtual Rigid Body (VRB) [3] can preserve prescribed spatial patterns in structured environments, but they are prone to local minima, oscillations, and deadlock in the presence of obstacles, and their reliance on handcrafted potential functions limits adaptability in dynamic scenarios. Model Predictive Control (MPC) [4] improves flexibility by optimizing over finite horizons and explicitly considering system dynamics, but its dependence on accurate modeling and heavy computational demand restricts real-time deployment and scalability to large swarms. These limitations make it difficult for traditional approaches to guarantee reliable performance when both agile target tracking and formation maintenance must be achieved in cluttered environments. In contrast, Multi-Agent Reinforcement Learning (MARL) [5], [6] offers a data-driven alternative that can learn cooperative behaviors directly from interaction with the environment. By adapting to uncertainties and complex dynamics, MARL has the potential to overcome some of the rigidity inherent in classical methods. Nevertheless, most existing MARL frameworks still face challenges in scalability when applied to large groups of agents, and often lack explicit mechanisms to enforce geometric formation constraints.



(a) $N = 12$



(b) $N = 18$

Fig. 1: Demonstration of UAV swarm formation. The swarm tracks a moving target (red dot with dashed trajectory) while forming the letters "ICRA". Gray cylinders denote obstacles. The colored curves represent UAV trajectories, with the color gradient indicating temporal evolution. The horizontal bar depicts time progression.

This work presents a unified MARL framework that integrates graph-based representation learning with geometric formation constraints to achieve scalable and robust UAV swarm coordination for simultaneous formation control and target tracking. The main contributions are:

- A conflict-aware graph representation that captures both local interactions and global formation geometry for robust swarm coordination.
- A hierarchical policy combining motion primitive selection with continuous refinement to generate agile and feasible maneuvers.
- A unified framework that jointly addresses formation control and target tracking in cluttered environments.
- Extensive validation in simulations and real-world experiments, demonstrating scalability and robustness.

[1]School of Electronics, Peking University, Beijing 100871, China.
Correspondence: wanghaowen19@stu.pku.edu.cn

II. Related Work

A. Target Tracking and Formation Control

In multi-UAV systems, achieving reliable target tracking while preserving formation is a fundamental but highly challenging problem, particularly in cluttered environments. Target tracking requires the swarm to continuously adjust its collective trajectory in accordance with the target, which may involve rapid or irregular variations in velocity and direction [7]. At the same time, formation control imposes geometric constraints among UAVs to ensure sensing coverage, communication efficiency, and collision avoidance [8]. The simultaneous pursuit of these two objectives introduces intrinsic conflicts: compact formations may restrict maneuverability in obstacle-rich regions, whereas sparse formations often compromise coordination and sensing effectiveness.

Classical formation control methods, such as APF [9], [2] and VRB [3] are capable of maintaining prescribed spatial patterns in structured environments, but they frequently encounter local minima and deadlock in the presence of obstacles. MPC frameworks provide improved adaptability by optimizing control sequences over a finite horizon, and distributed MPC variants have been applied to multi-UAV formations [10]. However, their dependence on accurate models and the computational burden of repeated constrained optimization limit scalability to large swarms and constrain real-time applicability. Furthermore, formation transitions in many existing strategies rely on pre-computed trajectories or fixed behavior rules, which can reduce flexibility when facing unforeseen target maneuvers and cluttered environments.

Traditional methods struggle to maintain stable formations and follow targets flexibly in complex environments. This highlights the need for more adaptive frameworks that combine geometric constraints with responsive pursuit.

B. MARL for UAV Control

MARL has emerged as a powerful framework for enabling coordinated control in UAV swarms by unifying perception, decision-making, and control [11], [12]. A central challenge in this setting is how agents exchange and aggregate information to achieve efficient cooperation. Predefined communication structures such as ring, grid, or leader–follower topologies [13] offer stable and computationally efficient solutions, but they are often limited in adapting to complex coordination patterns. More recently, attention-based mechanisms [14] have been introduced to enable UAVs to adaptively weigh the importance of neighboring agents, leading to more flexible coordination strategies.

GNNs provide a natural extension by modeling multi-agent systems as graphs, where nodes represent UAVs and edges capture their spatial or communication relationships [15], [16]. This representation allows scalable information aggregation through message passing, enabling UAVs to reason about both local and higher-order interactions. These advances have shown significant potential for large-scale swarm control. Nonetheless, most existing MARL approaches for UAVs emphasize interaction modeling but lack explicit incorporation of geometric formation constraints, which are essential for tasks requiring both precise formation maintenance and agile target tracking.

III. System Model

A. Framework Overview

The proposed framework addresses the problem of simultaneous UAV formation control and target tracking in complex, obstacle-rich environments by unifying perception, representation, and control into a graph-based MARL pipeline. The framework is shown in Fig. 2. Each UAV perceives its surrounding 3D environment through a voxel-based obstacle encoding module, which compresses raw occupancy information into a compact hidden state. This local feature is combined with target-relative measurements and formation guidance gradients to form the agent's observation. To capture inter-agent dependencies, observations are structured as a graph where nodes represent UAVs and edges encode relative spatial relationships. A graph neural network with attention-based message passing aggregates information from neighboring agents, enabling scalable reasoning about both local interactions and global formation geometry.

On top of this representation, each UAV learns a hierarchical policy within a parameterized action space. Specifically, a discrete head selects motion primitives from a predefined library, while a continuous refinement head adjusts trajectory offsets and dynamic constraints, producing dynamically feasible short-term maneuvers. Training is performed under a centralized training with decentralized execution (CTDE) paradigm using Multi-Agent Proximal Policy Optimization (MAPPO) [17]. A composite reward function integrates formation maintenance, target tracking accuracy, obstacle avoidance, and trajectory smoothness, ensuring both safety and coordination. This end-to-end framework allows UAV swarms to maintain stable formations and track targets robustly across varying scales and environmental complexities.

B. Performance Metrics

Inspired by [18], we adopt a graph representation to represent UAV formation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V} = \{1, 2, \ldots, N\}$ is the set of vertices representing UAVs and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges representing the distances between UAVs. The adjacency matrix $\mathbf{M}$ and degree matrix $\mathbf{D}$ can be defined. And the symmetric normalized Laplacian matrix is computed as:

$$\hat{\mathbf{L}} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{M})\mathbf{D}^{-1/2}. \tag{1}$$

Given a desired formation with the Laplacian matrix $\hat{\mathbf{L}}_{\text{des}}$, we quantify the error between the current and desired formations using the Frobenius norm of their difference:

$$f = \|\hat{\mathbf{L}} - \hat{\mathbf{L}}_{\text{des}}\|_F^2. \tag{2}$$

The overall target tracking performance is measured by the Euclidean distance between the swarm centroid $\bar{\mathbf{p}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i$ and the target position $\mathbf{p}_0$, where a smaller distance indicates more precise tracking.

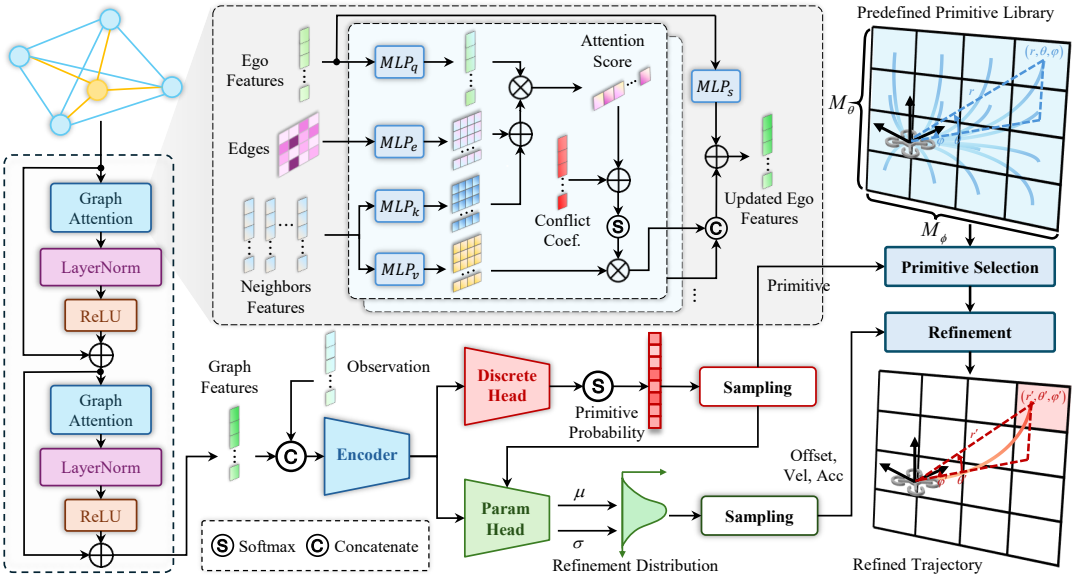


Fig. 2: Overview of the proposed framework. Each UAV encodes local observations and obstacle features into graph representations, where conflict-aware attention aggregates neighborhood information. A hierarchical policy then selects a motion primitive from a predefined library and refines it with continuous offsets, velocity, and acceleration, yielding a refinement distribution. By sampling from both heads, the agent generates dynamically feasible short-term maneuvers that combine high-level decision-making with fine-grained control.

C. Motion Primitive Library Definition

The motion primitive library is a structured set of trajectory templates used for efficient trajectory generation in motion planning [1]. It discretizes the space of possible movement directions in three dimensions, forming a set of short-duration, dynamically feasible trajectory candidates. Specifically, the library is built by discretizing the motion space using spherical coordinates. A fixed planning radius d is defined, and uniform sampling is performed over the elevation angle θ and the azimuth angle ϕ, forming a $M_\phi \times M_\theta$ directional grid, as shown in the right top part of Fig. 2. Each grid cell corresponds to a motion primitive whose endpoint position is given by:

$$\mathbf{q}_{mn} = (d\cos\theta_m \cos\phi_n, d\cos\theta_m \sin\phi_n, d\sin\theta_m), \quad (3)$$

where θ_m and ϕ_n represent the sampled elevation and azimuth angles at the m-th row and n-th column of the grid.

The motion primitives can be refined using RL. RL actor predicts three offset values: Δd_{mn}, $\Delta\theta_{mn}$, and $\Delta\phi_{mn}$, which adjust the original distance and direction to produce a better-suited target position $\mathbf{q}'_{mn}$. Additionally, the network also predicts terminal velocity $\mathbf{v}_{mn}$ and acceleration $\mathbf{a}_{mn}$ to obtain complete trajectory constraints, which include $[\mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i, \mathbf{p}_i + \mathbf{q}'_{mn}, \mathbf{v}_{mn}, \mathbf{a}_{mn}]$ representing the position, velocity, and acceleration of the trajectory at both the initial and terminal states, respectively. These boundary states can be transformed into the coefficients of a fifth-order polynomial, thereby constructing the trajectory of the UAV [19].

IV. RL Formulation and Training

A. Decision Process Formulation

We model the multi-agent cooperative control problem as a Decentralized Partially Observable Markov Decision Process, defined by the tuple $\langle N, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{G}, P, R, \gamma\rangle$. Here, N denotes the number of agents, $\mathcal{S}$ is the global state space, and $\mathcal{O}$ is the observation space. The action space $\mathcal{A}$ is defined as parameterized. Action of agent i is therefore written as $\mathbf{a}_i(k) = \big(a_i^{\mathrm{d}}(k), \mathbf{a}_i^{\mathrm{p}}(k)\big) \in \mathcal{A}_i$, where $a_i^{\mathrm{d}}(k) \in \mathcal{A}_i^{\mathrm{d}}$ is a discrete action and $\mathbf{a}_i^{\mathrm{p}}(k) \in \mathcal{A}_i^{\mathrm{p}}$ is a continuous parameter. The term $\mathcal{G} = \{\mathcal{G}_1(k), \ldots, \mathcal{G}_N(k)\}$ denotes a set of agent-centric graphs. The transition function $P(\mathbf{s}(k+1) \mid \mathbf{s}(k), \mathbf{A}(k))$ defines the probability of transitioning from state $\mathbf{s}(k)$ to $\mathbf{s}(k+1)$ given the joint action $\mathbf{A}(k) = [\mathbf{a}_1(k), \ldots, \mathbf{a}_N(k)]$. The reward function $R(\mathbf{s}(k), \mathbf{A}(k))$ assigns a feedback based on the current state and joint action. The discount factor γ determines the relative importance of future rewards. Each agent seeks to learn a hierarchical stochastic policy:

$$\begin{aligned}\pi_\psi\big(\mathbf{a}_i(k) \mid \mathbf{o}_i(k), \mathcal{G}_i(k)\big) = \pi_\psi^{\mathrm{d}}\big(a_i^{\mathrm{d}} \mid \mathbf{o}_i(k), \mathcal{G}_i(k)\big) \\ \cdot \pi_\psi^{\mathrm{p}}\big(\mathbf{a}_i^{\mathrm{p}} \mid \mathbf{o}_i(k), \mathcal{G}_i(k), a_i^{\mathrm{d}}\big),\end{aligned} \quad (4)$$

where $\mathbf{o}_i(k)$ is the observation, π_ψ^{d} selects a discrete action, and π_ψ^{p} determines the continuous parameters. For notational simplicity, we omit time index k in the following sections.

B. Voxel Feature Extraction

We designed a lightweight network model to extract forward occupancy grid information of the UAV for obstacle avoidance, as shown in Fig. 3. Here, the x-axis aligns with the UAV's heading direction, while the y and z axes span the lateral and vertical dimensions, respectively. The voxel map covers the $5\,m \times 10\,m \times 5\,m$ volume directly in front of the UAV, discretized at a resolution of $0.1\,m$ along each axis. Instead of processing the full 3D voxel volume directly, we project the map along the x-axis to construct a structured, three-channel feature map on the yz-plane. Each channel encodes a distinct aspect of the spatial layout: (1) *Nearest-Obstacle Depth* captures the normalized distance to the first occupied voxel along the forward direction; (2) *Occupancy*

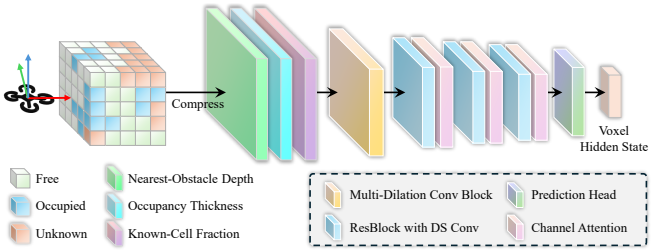


Fig. 3: Overview of the obstacle feature extraction and encoding.

Thickness quantifies how thick or extended the obstacle is within that column of space; (3) *Known-Cell Fraction* measures the proportion of voxels in the column that have been observed, helping to disambiguate between free and unobserved regions. These features are normalized to the range $[0, 1]$ and stacked to form a 2D input.

We then apply a lightweight convolutional encoder composed of a multi-dilation convolutional block followed by three stages of depthwise–separable residual blocks, each reducing spatial resolution while increasing channel capacity. After each block, a standalone channel attention module is applied to adaptively recalibrate channel-wise features. A prediction head consisting of a global average pooling layer and a fully connected layer produces a compact one-dimensional hidden state $\mathbf{m}$ that summarizes the spatial obstacle structure in front of the UAV.

C. Graph-Based Representation

Each UAV maintains an observation vector as $\mathbf{o}_i = [\mathbf{v}_i,\ \mathbf{p}_{i0},\ \mathbf{v}_{i0},\ \nabla f_i,\ \mathbf{m}_i,\ \mathbf{a}_i(k-1)]$, where $\mathbf{v}_i$ is the UAV's velocity, $\mathbf{p}_{i0}$ denotes the estimated relative position from UAV to the target, $\mathbf{v}_{i0}$ is the velocity of UAV with respect to the target, ∇f_i is the gradient of the formation error with respect to the UAV's position, $\mathbf{a}_i(k-1)$ is the action of the previous time step, and $\mathbf{m}_i$ is the local voxel information.

Each UAV constructs an agent observation graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, centered on itself. The node set $\mathcal{V}_i$ includes UAV i itself and other UAVs, while directed edges $e_{ij} \in \mathcal{E}_i$ indicate distance from UAV i to node j. Each edge carries a feature representing the Euclidean distance $||\mathbf{p}_j - \mathbf{p}_i||_2$ between the two UAVs. For each node $j \in \mathcal{V}_i$, a feature vector is defined to encode the spatial information as $\mathbf{f}_j = [\mathbf{p}_{ij}, \mathbf{v}_j, \nabla f_j, \kappa_j]$, where $\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i$ is the relative position of UAV j with respect to UAV i, $\mathbf{v}_j$ denotes the velocity of UAV j, and κ_j is the conflict intensity coefficient, which is defined as the cosine similarity between formation gradient and the relative target velocity vector as:

$$\kappa_j = \frac{1}{2}\left(1 - \frac{\mathbf{v}_{j0} \cdot \nabla f_j}{\|\mathbf{v}_{j0}\|\ \|\nabla f_j\|}\right). \tag{5}$$

The parameter κ_j is employed to quantify conflict arising from target tracking and formation maintenance. A larger divergence among these directional vectors leads to a greater value of κ, indicating a higher level of conflict.

D. Information Aggregation

In the proposed framework, the information aggregation module aims to infer information about the neighborhood of each agent using a GNN with a message passing framework. For each agent, its node features $\mathbf{f}_i$ are updated as follows:

$$\mathbf{f}_i' = \mathbf{W}_s \mathbf{f}_i + \sum \alpha_{i,j} \mathbf{W}_v \mathbf{f}_j, \tag{6}$$

where $\mathbf{W}$ are learnable weight matrices. The attention coefficients $\alpha_{i,j}$ are computed via a multi-head dot product attention mechanism, allowing agents to selectively prioritize messages from their neighbors according to their importance:

$$\alpha_{i,j} = \text{softmax}\left(\frac{(\mathbf{W}_q \mathbf{f}_i)^\top (\mathbf{W}_k \mathbf{f}_j + \mathbf{W}_e e_{ij})}{\sqrt{c}} + \beta \kappa_j\right), \tag{7}$$

where e_{ij} are the distances between UAV i and j, c is the output dimension for that specific layer, and β controls the influence of conflict intensity coefficient on the attention weights. The term $\beta\kappa_j$ directly increases the attention weight of high-conflict nodes, enabling the information aggregation process to focus more on critically disturbed agents. As a result, neighboring agents will proactively adjust the formation structure around these high-conflict nodes.

By stacking multiple such message-passing layers, information can be propagated between agents that are higher-order neighbors. For each agent, this module aggregates information from the neighboring nodes into a fixed-sized vector $\mathbf{n}_i$. This aggregated vector is then concatenated with the agent's own observation $\mathbf{o}_i$ to form an enhanced state representation $\mathbf{h}_i = [\mathbf{o}_i, \mathbf{n}_i]$. This architecture allows the method to dynamically adapt to a changing number of agents in the environment while remaining invariant to the permutation of the observed agents.

To support CTDE framework, the model also incorporates a global graph-level aggregation mechanism during training. This mechanism applies a mean pooling operation across all agent embeddings. The resulting vector, which has a fixed size regardless of the number of agents, is passed to the critic network to evaluate joint state-action values.

E. Action and Reward Design

In our system each agent executes an action in a parameterized form $\mathbf{a}_i = \left(a_i^{\mathrm{d}}, \mathbf{a}_i^{\mathrm{p}}\right)$, which contains a discrete choice and a continuous refinement. The discrete component a_i^{d} selects a motion primitive from the predefined primitive library, where each primitive corresponds to a short-horizon maneuver represented by a sampled direction in spherical coordinates. The continuous component $\mathbf{a}_i^{\mathrm{p}}$ further adjusts this primitive by modifying offsets $(\Delta d, \Delta\theta, \Delta\phi)$ and by specifying terminal velocity and acceleration. By combining the two parts, the action provides a description of a dynamically feasible short-term trajectory that integrates high-level maneuver selection with low-level control refinement [1].

We design a composite reward function consisting of multiple components, each addressing a key objective. The total reward for a single agent is defined as:

$$r = \lambda_f r_f + \lambda_t r_t + \lambda_o r_o + \lambda_s r_s, \tag{8}$$

where $\lambda_f, \lambda_t, \lambda_o, \lambda_s$ are the weighting coefficients.

The formation maintenance reward r_f measures how close the current formation is to the desired configuration. We

evaluate this term by sampling the trajectory at discrete time intervals and summing the formation error as $r_f = -\sum_0^{T/\delta t} f \cdot \delta t$, where T is the trajectory duration.

The target tracking reward r_t encourages the formation to stay close to the target by penalizing the distance between the UAV swarm's centroid $\bar{\mathbf{p}}$ and the target position $\mathbf{p}_0$, defined as $r_t = -\sum_0^{T/\delta t} \|\bar{\mathbf{p}} - \mathbf{p}_0\|_2^2 \cdot \delta t$.

The safety is handled by the obstacle avoidance reward r_o, which penalizes trajectories that pass close to obstacles. This term is defined as $r_o = -\sum_0^{T/\delta t} \exp\left(-d(\mathbf{p}) + d_0\right) \cdot \delta t$, where $d(\cdot)$ is the Euclidean Signed Distance Field (ESDF) query. The d_0 controls the safety margin.

The smoothness reward r_s enforces dynamic feasibility by minimizing the high-order derivatives of the generated trajectory. We penalize the fourth-order derivative of position with respect to time using the widely used minimum-snap formulation [20] as $r_s = -\sum_0^{T/\delta t} \left\| \left(d^4\mathbf{p}/dt^4\right) \right\|^2 \cdot \delta t$.

F. Policy Training

To train the proposed policy, we adopt a CTDE paradigm based on MAPPO [21]. Each agent employs a shared backbone to encode local observations and graph-based neighborhood information. On top of this representation, the actor network contains two components: (a) a discrete head that outputs a categorical distribution over motion primitives, and (b) a parameter head that refines the chosen primitive with continuous parameters. To ensure scalability, the parameter prediction is implemented via a shared conditional module: the selected primitive is embedded and concatenated with the backbone features, and a lightweight MLP generates the corresponding Gaussian distribution parameters. During execution, the agent first samples a primitive and subsequently its parameters, thus generating a trajectory with appropriate motion direction and dynamic feasibility.

The overall log-probability of an action is factorized as

$$\begin{aligned} \log \pi_\psi(\mathbf{a}_i \mid \mathbf{o}_i, \mathcal{G}_i) = \; & \log \pi_\psi^{\mathrm{d}}(a_i^{\mathrm{d}} \mid \mathbf{o}_i, \mathcal{G}_i) \\ & + \log \pi_\psi^{\mathrm{p}}(\mathbf{a}_i^{\mathrm{p}} \mid \mathbf{o}_i, \mathcal{G}_i, a_i^{\mathrm{d}}). \end{aligned} \tag{9}$$

MAPPO employs this factorization to compute the clipped surrogate objective with generalized advantage estimation (GAE) [21]. Entropy regularization is applied to both discrete and continuous components to encourage exploration. The critic, operating on pooled global graph features, estimates the state value following the standard MAPPO design.

V. Experiments

A. Experimental Setup

We first provide detailed specifications of the key parameters used in our framework. The motion primitive library was discretized into a directional grid of size $M_\phi \times M_\theta = 5 \times 3$, and the planning radius was fixed at $d = 5\,m$. For the attention weight computation, the conflict intensity coefficient was set to $\beta = 0.5$, which was selected through grid search based on validation performance. The composite reward function employed empirically tuned weights to balance multiple objectives: formation maintenance $\lambda_f = 1.0$, target tracking $\lambda_t = 1.5$, obstacle avoidance $\lambda_o = 2.0$, and trajectory smoothness $\lambda_s = 0.1$. Training was performed using the MAPPO algorithm with a discount factor of $\gamma = 0.99$ and a generalized advantage estimation (GAE) parameter of $\tau = 0.95$. Both the policy and value networks were optimized using Adam with a learning rate of 3×10^{-4}. The PPO clipping parameter was fixed at $\epsilon = 0.2$.

B. Comparative Analysis of Formation Tracking

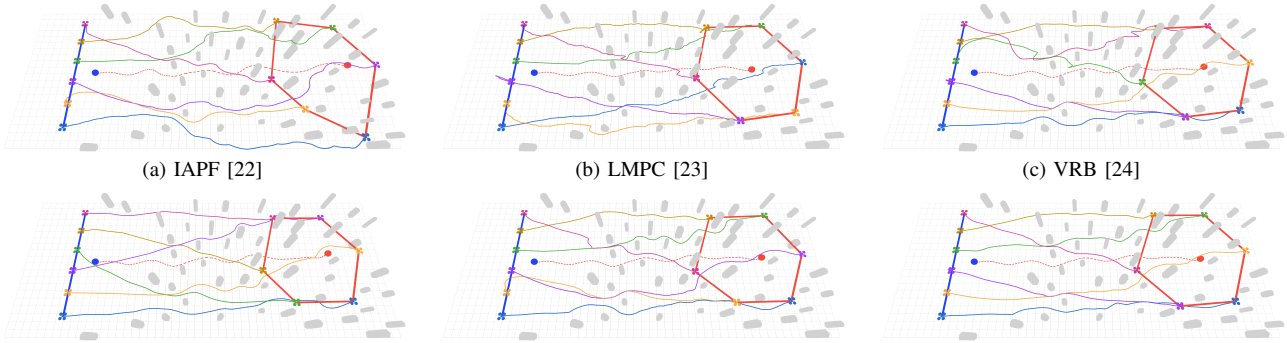
We compared our method with state-of-the-art algorithms for multi-UAV formation control and tracking, including IAPF [22], LMPC [23], VRB [24], Swarm-Formation [18], and FARO-RL [25]. For fairness, a target estimation and tracking module was added to all baselines. Simulations were run in a $100\,m \times 40\,m$ environment with 6 UAVs under low (50), medium (75), and high (100) obstacle densities. The target moved along the x-axis at $1\,m/s$ with lateral sinusoidal motion, while the swarm maintained a circular formation. Fig. 4 shows trajectories for each algorithm, and Table I reports quantitative results. IAPF and VRB often fall into local minima, LMPC produces oscillatory paths, and Swarm-Formation relies on rigid relative position constraints that break down in cluttered environments. Learning based method FARO-RL fails to preserve formation geometry due to the lack of topological modeling. In contrast, our method produces smoother trajectories and more stable formations. Across all densities, it achieves the lowest tracking error and collision rate while maintaining the best formation. Its advantage stems from the conflict-aware graph balances local and global coordination, combined with a parameterized action space for primitive selection and refinement, ensuring robust tracking and stable formation in cluttered environments.

C. MARL Algorithms Comparison

In this subsection, we evaluated multiple MARL algorithms on the task, including centralized methods such as MADDPG [5], RMADDPG [6], and RMAPPO [21], as well as graph-based methods such as GPG [15], DGN [16], and EMP [26]. We tested each algorithm with 4, 8, and 12 agents under 2×10^6 training steps and repeated each setup 10 times. The training curves are shown in Fig. 5, while Table II reports the quantitative results in terms of cumulative reward, collision rate, tracking error, and formation error. Our method stands out due to its distinctive designs. First, the conflict-aware attention mechanism adaptively increases the weights of high-conflict neighbors during message passing, ensuring that agents focus on resolving critical coordination issues between target tracking and formation maintenance. Moreover, the parameterized action space combines discrete motion primitive selection with continuous refinement, enabling UAVs to generate dynamically feasible maneuvers that balance agility with stability.

D. Formation Definition Evaluation

We evaluated four formation definition methods: Position, Displacement, Distance, and Graph. The UAVs were required to switch between predefined formations while tracking the



(a) IAPF [22] (b) LMPC [23] (c) VRB [24]

(d) Swarm-Formation [18] (e) FARO-RL [25] (f) Ours

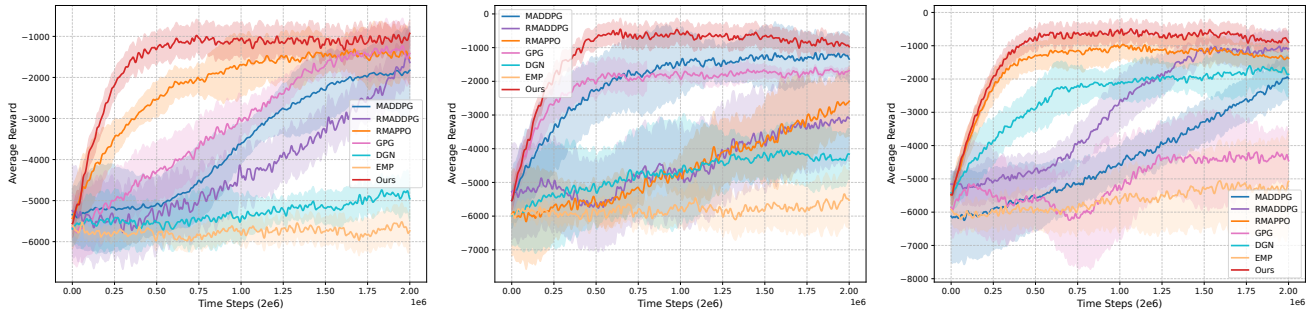
Fig. 4: Visualization of UAV and target trajectories for different algorithms in the low-density obstacle scenario. The colored lines represent the paths of the 6 UAVs, while the red dashed line indicates the target's trajectory. Gray cylinders denote obstacles.

TABLE I: Performance Comparison Across Different Scenarios and Algorithms

Scenario	Algorithm	Tra. Err. (m)	Col. Rate (%)	Obs. Dist. (m)	UAV Dist. (m)	For. Err. (%)
Low Density	IAPF [22]	0.18	32.3	0.41	1.53	0.923
	LMPC [23]	0.15	31.9	0.54	1.61	0.847
	VRB [24]	0.22	22.6	0.63	1.66	0.826
	Swarm-Formation [18]	0.19	**1.3**	**1.51**	**2.04**	0.214
	FARO-RL [25]	0.14	1.6	1.12	1.90	0.185
	Ours	**0.11**	1.4	1.24	1.96	**0.092**
Medium Density	IAPF [22]	0.21	35.2	0.36	1.56	1.157
	LMPC [23]	0.17	34.8	0.41	1.59	0.973
	VRB [24]	0.26	25.5	0.52	1.61	0.841
	Swarm-Formation [18]	0.23	3.1	1.07	1.79	0.183
	FARO-RL [25]	0.16	2.4	1.16	1.86	0.126
	Ours	**0.13**	**2.2**	**1.20**	**1.92**	**0.103**
High Density	IAPF [22]	0.25	38.3	0.27	1.41	1.584
	LMPC [23]	0.20	37.7	0.32	1.49	0.916
	VRB [24]	0.31	28.8	0.39	1.56	0.852
	Swarm-Formation [18]	0.27	4.1	0.77	1.71	0.248
	FARO-RL [25]	0.19	3.2	0.86	1.83	0.151
	Ours	**0.16**	**2.9**	**1.05**	**1.88**	**0.115**

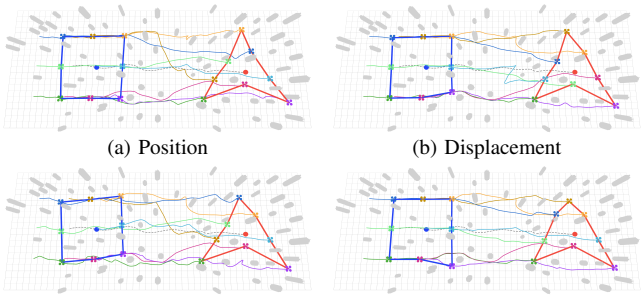
TABLE II: Performance Comparison Across Different MARL Methods for Varying UAV Numbers

Method	$N = 4$				$N = 8$				$N = 12$			
	Reward	Col. Rate (%)	Tra. Err. (m)	For. Err. (%)	Reward	Col. Rate (%)	Tra. Err. (m)	For. Err. (%)	Reward	Col. Rate (%)	Tra. Err. (m)	For. Err. (%)
MADDPG [5]	-1828.7	7.8	0.24	0.44	-1337.9	8.5	0.31	0.56	-1980.9	9.2	0.42	0.76
RMADDPG [6]	-1636.4	6.8	0.21	0.37	-3097.1	7.2	0.28	0.50	-1090.2	8.4	0.38	0.68
RMAPPO [21]	-1528.4	5.5	**0.11**	0.33	-2593.0	6.3	0.25	0.40	-1378.3	7.1	0.35	**0.21**
GPG [15]	-1417.7	6.2	0.19	0.35	-1683.4	6.7	0.27	0.52	-4452.0	8.5	0.39	0.70
DGN [16]	-4954.0	7.0	0.22	0.41	-4160.1	8.2	0.29	0.54	-1857.5	9.0	0.36	0.69
EMP [26]	-5736.5	7.5	0.23	0.45	-5529.0	8.0	0.28	0.44	-5075.3	8.9	0.34	0.64
Ours	**-926.4**	**1.8**	0.12	**0.10**	**-973.4**	**2.2**	**0.15**	**0.13**	**-895.3**	**4.9**	**0.18**	0.23



(a) $N = 4$ (b) $N = 8$ (c) $N = 12$

Fig. 5: Training curves of different MARL algorithms with varying numbers of agents. The solid lines denote the average episodic reward over 10 independent runs, while the shaded regions represent the variance across runs.



(a) Position (b) Displacement

(c) Distance (d) Graph

Fig. 6: Trajectory visualization of formation switching from square to arrow under different definition methods.

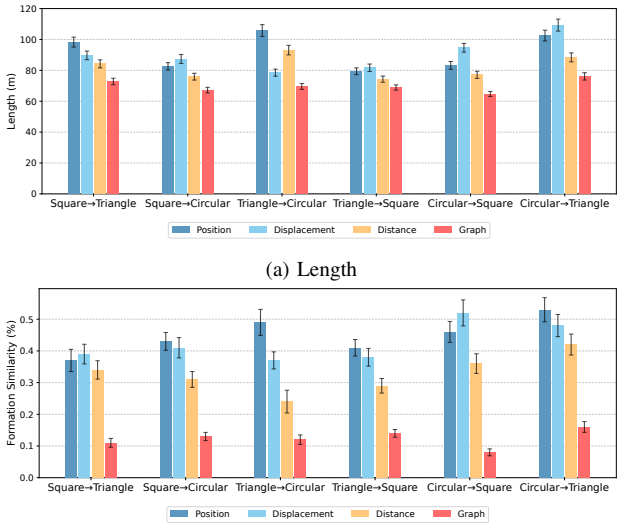
moving target. The position-based method fixed absolute UAV positions relative to the target, while the displacement- and distance-based methods constrained relative displacement vectors or inter-UAV distances. In contrast, the graph-based method leveraged structural relationships without prescribing absolute or relative coordinates. As shown in Fig. 6, which illustrates a formation transition from a square to an arrow shape, the graph-based definition produces smoother and more coordinated trajectories, enabling the swarm to reconfigure formations efficiently while maintaining stability. In comparison, the other three methods incur redundant maneuvers and slower convergence due to rigid positional constraints. Quantitative results in Fig. 7 further demonstrate that the graph-based definition consistently achieves better formation maintenance and shorter path length, demonstrating superior adaptability in cluttered environments.

E. Generalization Across Varying UAV Numbers

We conducted a series of cross-scale experiments in which the number of UAVs used during training differed from that used during testing. The results are shown in Table III. The results demonstrate that our method maintains consistent and robust performance across varying swarm sizes. Notably, policies trained with smaller swarms remained effective when directly applied to larger swarms, demonstrating that the learned strategies scale effectively without retraining. The success of this generalization is attributed to the graph-based design and decentralized policy learning framework,

TABLE III: Performance across Training and Testing UAV Numbers

Test \ Train		$N = 4$	$N = 8$	$N = 12$
$N' = 4$	Col. Rate (%)	1.8	1.6	1.8
	Tra. Err. (m)	0.12	0.14	0.16
	For. Err. (%)	0.10	0.14	0.17
$N' = 8$	Col. Rate (%)	2.2	2.2	2.5
	Tra. Err. (m)	0.16	0.15	0.16
	For. Err. (%)	0.12	0.13	0.20
$N' = 12$	Col. Rate (%)	5.2	5.0	4.9
	Tra. Err. (m)	0.20	0.19	0.18
	For. Err. (%)	0.28	0.26	0.23
$N' = 16$	Col. Rate (%)	5.8	5.6	5.4
	Tra. Err. (m)	0.22	0.21	0.20
	For. Err. (%)	0.22	0.27	0.25



(a) Length

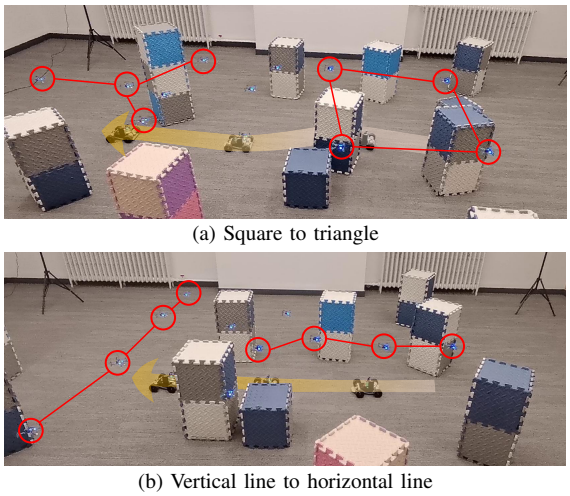
(b) Formation Error

Fig. 7: Quantitative comparison of different formation definition methods in terms of total path length and formation error.

which enables each UAV to operate autonomously while still achieving coordinated global behavior. As shown in Fig. 1, the framework further scaled to swarms of 12 and 18 UAVs that successfully tracked a moving target while forming the letters "ICRA", confirming its strong scalability and adaptability across varying swarm sizes.

F. Real-World Validation

We conducted real-world experiments to validate the effectiveness of the proposed method based on the Prometheus [27] project. The experimental platform consisted of four Crazyflie UAVs, with precise localization of both the UAVs and the target provided by the HTC



(a) Square to triangle

(b) Vertical line to horizontal line

Fig. 8: Real-world experiments of UAV formation switching while tracking a ground vehicle in an obstacle-rich environment. Each subfigure shows three snapshots of the UAV formation and the target. The yellow arrow indicates the target's direction, while the red circles mark the UAVs within the formation at the timestamps.

Vive Lighthouse system. In the experimental environment, various obstacles were deployed, and the AMOVLAB's P600 quadrotor equipped with Livox MID-360 was used to map the environment and construct a voxel occupancy grid. The UAVs were required to track the target while maintaining a predefined formation and adaptively reconfiguring the formation during flight. A laptop with an Intel i9-14900HX CPU and NVIDIA GeForce RTX 5060 Laptop GPU served as the ground station. We configured multiple ROS nodes to represent different UAVs, each independently computing its own trajectory using the proposed algorithm and broadcasting it to the corresponding UAV for execution. The average planning time for the UAV trajectory was $2.13\,ms$. Two formation transition scenarios were examined: (a) square to triangle and (b) vertical line to horizontal line. The results in Fig. 8 demonstrated that the UAV swarm successfully completed formation reconfiguration while consistently tracking the target through cluttered environments. The UAVs exhibited smooth and coordinated maneuvers, maintaining stable inter-agent spacing and promptly adapting their positions to preserve the overall formation.

VI. Conclusion

This paper introduces a graph-based MARL framework for UAV formation control and target tracking. The framework employs a conflict-aware graph representation to capture both local interactions and global formation geometry, while a hierarchical policy combines motion primitive selection with continuous refinement to generate dynamically feasible maneuvers in cluttered environments. To enhance real-world applicability, a lightweight voxel-based obstacle encoding module is integrated, enabling efficient perception of environments without incurring heavy computational costs. Extensive simulations showed that the proposed method outperformed classical control and MARL baselines, achieving lower tracking error, fewer collisions, and better formation across varying obstacle densities. Moreover, policies learned with smaller swarms can be directly applied to larger ones, confirming strong scalability across varying UAV numbers. Real-world experiments further validated its practicality.

References

[1] H. Wang, S. Zhang, Y. Sun, Z. Wang, J. Sun, and B. Zhu, "Swift: A distributed one-stage planner for efficient multi-quadrotor trajectory optimization," *IEEE Transactions on Automation Science and Engineering*, pp. 1–1, 2025.

[2] Z. Pan, C. Zhang, Y. Xia, H. Xiong, and X. Shao, "An improved artificial potential field method for path planning and formation control of the multi-uav systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1129–1133, 2022.

[3] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916–923, 2018.

[4] Z. Du, H. Zhang, Z. Wang, and H. Yan, "Model predictive formation tracking-containment control for multi-uavs with obstacle avoidance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 6, pp. 3404–3414, 2024.

[5] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017.

[6] R. E. Wang, M. Everett, and J. P. How, "R-maddpg for partially observable environments and limited communication," 2020.

[7] P. Wu, Y. Li, and D. Xue, "Uav target tracking: a survey," *Artificial Intelligence Review*, vol. 58, no. 11, pp. 1–62, 2025.

[8] Y. Bu, Y. Yan, and Y. Yang, "Advancement challenges in uav swarm formation control: A comprehensive review," *Drones*, vol. 8, no. 7, p. 320, 2024.

[9] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 1398–1404 vol.2, 1991.

[10] M. Yang, X. Guan, M. Shi, B. Li, C. Wei, and K.-F. C. Yiu, "Distributed model predictive formation control for uavs and cooperative capability evaluation of swarm," *Drones*, vol. 9, no. 5, 2025.

[11] R. Mousavifard, K. Alipour, M. A. Najafqolian, and P. Zarafshan, "Formation control of multi-quadrotors based on deep q-learning," in *2022 10th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pp. 172–177, 2022.

[12] W. Wang, L. Wang, J. Wu, X. Tao, and H. Wu, "Oracle-guided deep reinforcement learning for large-scale multi-uavs flocking and navigation," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10280–10292, 2022.

[13] G. Raja, S. Essaky, A. Ganapathisubramaniyan, and Y. Baskar, "Nexus of deep reinforcement learning and leader–follower approach for aiot enabled aerial networks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 8, pp. 9165–9172, 2023.

[14] J. Wu, D. Li, Y. Yu, L. Gao, J. Wu, and G. Han, "An attention mechanism and adaptive accuracy triple-dependent maddpg formation control method for hybrid uavs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 11648–11663, 2024.

[15] A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar, "Graph policy gradients for large scale robot control," in *Conference on robot learning*, pp. 823–834, PMLR, 2020.

[16] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," *Advances in neural information processing systems*, vol. 31, 2018.

[17] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, (Red Hook, NY, USA), Curran Associates Inc., 2022.

[18] L. Quan, L. Yin, C. Xu, and F. Gao, "Distributed swarm trajectory optimization for formation flight in dense environments," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4979–4985, IEEE, 2022.

[19] M. Lu, X. Fan, H. Chen, and P. Lu, "Fapp: Fast and adaptive perception and planning for uavs in dynamic cluttered environments," *IEEE Transactions on Robotics*, vol. 41, pp. 871–886, 2025.

[20] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: Springer Tracts in Advanced Robotics*, pp. 649–666, 2016.

[21] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[22] Y. Li, P. Zhang, Z. Wang, D. Rong, M. Niu, and C. Liu, "Multi-uav obstacle avoidance and formation control in unknown environments," *Drones*, vol. 8, no. 12, p. 714, 2024.

[23] Z. Du, H. Zhang, Z. Wang, and H. Yan, "Model predictive formation tracking-containment control for multi-uavs with obstacle avoidance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 6, pp. 3404–3414, 2024.

[24] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916–923, 2018.

[25] Y. Xie, C. Yu, H. Zang, F. Gao, W. Tang, J. Huang, J. Chen, B. Xu, Y. Wu, and Y. Wang, "Multi-uav formation control with static and dynamic obstacle avoidance via reinforcement learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20410–20417, IEEE, 2025.

[26] A. Agarwal, S. Kumar, K. Sycara, and M. Lewis, "Learning transferable cooperative behavior in multi-agent teams," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, (Richland, SC), p. 1741–1743, International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[27] Amovlab, "Prometheus autonomous uav opensource project." https://github.com/amov-lab/Prometheus, 2020.