

Article

Hierarchical Graph-Attention Multi-Agent Reinforcement Learning for Safe-Separation-and-Collision-Avoidance Coordination of Heterogeneous UAV Swarms

Xudong Zhang ¹, Junqiang Bai ¹, Kang Chen ^{1,*} and Xinzhuang Chen ²

¹ Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710129, China; zhangxudong2022@mail.nwpu.edu.cn (X.Z.); junqiang@nwpu.edu.cn (J.B.)

² College of Mathematics and Computer Science, Yan'an University, Yan'an 716000, China; chenxzh@yau.edu.cn

* Correspondence: kangchen@nwpu.edu.cn

Highlights

What are the main findings?

- HG-MARL achieves high task success (82.6–92.9%) across heterogeneous UAV swarm scenarios.
- HG-MARL improves task success over MAPPO by 15.1–21.4 percentage points, and all six HG-MARL-vs-MAPPO/QMIX comparisons yield $p < 0.01$.

What are the implications of the main findings?

- The framework supports civilian UAV tasks such as inspection, emergency response, and search-and-rescue.
- It offers a safety-aware benchmark for resilient UAV-swarm autonomy.

Abstract

Safe-separation-and-collision-avoidance unmanned aerial vehicle (UAV) swarms are increasingly used for inspection, emergency response, environmental monitoring, and search-and-rescue support in cluttered airspace where communication links may be delayed, degraded, or intermittently unavailable. These applications require heterogeneous vehicles to maintain situational awareness, allocate tasks, and avoid hazards under partial observability and changing team topology. To address these challenges, this paper proposes a Hierarchical Graph-Attention Multi-Agent Reinforcement Learning architecture (HG-MARL) for safe-separation-and-collision-avoidance heterogeneous UAV swarm coordination. The proposed framework decomposes the task into high-level resource allocation and low-level local-control execution, uses graph attention for changing swarm topology, and applies Transformer memory, action masking, potential-field reward shaping, and domain-randomized simulation training. In the multi-scenario simulation summaries, HG-MARL achieves 92.9%, 89.8%, and 82.6% task success in Scenarios A–C, respectively, improving upon MAPPO by 15.1, 21.4, and 20.1 percentage points. Summary-statistic Welch tests show that all six HG-MARL comparisons against MAPPO and QMIX yield $p < 0.01$ with large effect sizes. Fair-control, reward-sensitivity, communication-degradation, safety-ablation, training-stability, latency, and transfer-oriented stress tests further support the contributions of the integrated architecture. The validation scope is simulator-based, with platform-level flight/HIL evaluation discussed as future work. These results suggest that HG-MARL is a promising simulation-validated framework for civilian UAV swarm coordination in collision-and-separation-critical and communication-degraded environments.



Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 15 May 2026

Revised: 26 June 2026

Accepted: 30 June 2026

Published: 3 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Keywords: multi-agent reinforcement learning; heterogeneous UAV swarm; safe-separation-and-collision-avoidance coordination; graph attention network; Transformer; domain randomization

1. Introduction

1.1. Safe-Separation-and-Collision-Avoidance Background and Requirements

Rapid advances in unmanned systems are increasingly influencing public-safety, inspection, and environmental-monitoring workflows [1,2]. Owing to their low unit cost, deployment flexibility, and scalability, unmanned aerial vehicle (UAV) swarms have attracted growing attention for cooperative sensing and response in infrastructure corridors, post-disaster assessment, and communication-degraded outdoor environments [3,4]. Compared with single-platform deployment, heterogeneous UAV swarms can distribute sensing, relay, and response roles across multiple platforms, but they must do so over unstructured terrain, uncertain weather, intermittent communication, dynamic obstacles, and incomplete local observations.

Traditional UAV swarm coordination methods often rely on rule-based logic or artificial potential fields (APFs) [5,6]. Rule-based methods depend on expert-crafted if-then rules, such as fixed avoidance maneuvers triggered by proximity sensing, and therefore lack adaptability to environmental changes and uncertain observations. APF-based methods are effective for local obstacle avoidance, but they may become trapped in local optima in multi-objective dynamic tasks, leading to coordination degradation or task failure [7]. More importantly, these methods are not well suited to heterogeneous safe-separation-and-collision-avoidance applications, where probe, relay, and responder UAVs must coordinate through tightly coupled cross-role dependencies.

1.2. Opportunities and Challenges of Multi-Agent Reinforcement Learning (MARL)

Multi-agent reinforcement learning (MARL) provides a learning-based alternative for cooperative decision-making in complex multi-agent systems [2,8]. By optimizing policies through large-scale simulated interactions, MARL can learn cooperative behaviors that are difficult to specify using manually designed rules [9]. Representative approaches include value-decomposition methods, such as QMIX, which factorizes the joint action-value function under a monotonicity constraint [10], and policy-gradient methods, such as multi-agent proximal policy optimization (MAPPO) [9]. These methods have achieved strong performance in standard multi-agent benchmarks, including the StarCraft Multi-Agent Challenge (SMAC), demonstrating the potential of MARL for scalable cooperative decision-making.

However, applying MARL to safe-separation-and-collision-avoidance heterogeneous UAV swarms remains challenging due to the complexity of role coordination, dynamic communication conditions, and incomplete situational awareness. Three key challenges are particularly relevant to this study.

Curse of dimensionality and credit assignment. As the swarm size increases from several platforms to larger heterogeneous formations, the joint state-action space grows combinatorially [2,3]. In task-level cooperative settings, it is difficult to determine the contribution of each role, especially for supporting agents such as probes or communication relays, which may slow policy convergence and weaken coordination efficiency [2,8].

Dynamic communication topology. In communication-degraded environments, links may become degraded or fail due to distance, terrain blockage, congestion, or system faults [11]. Fixed-dimensional message concatenation is therefore poorly suited to variable-

size swarms and changing graph structures, making adaptive communication modeling necessary [12].

Partial observability and long-horizon memory. Each UAV has only a limited field of view and may temporarily lose observations because of terrain occlusion, sensor limitations, or delayed information exchange. Agents must therefore infer hazard states and teammate intentions from incomplete and delayed observations. Although long short-term memory (LSTM) and gated recurrent unit (GRU) networks are commonly used for temporal modeling, they may struggle to capture long-range spatiotemporal dependencies in extended cooperative tasks [13,14].

1.3. Contributions

We present a Hierarchical Graph-Attention Multi-Agent Reinforcement Learning framework (HG-MARL) tailored to safe-separation-and-collision-avoidance heterogeneous UAV swarm coordination. The framework integrates hierarchical decision-making, graph-attention-based communication, Transformer-enhanced temporal representation, and safety-aware training mechanisms. The main contributions are summarized as follows: Explicit novelty statement.

The novelty of HG-MARL lies in a reproducible integration for heterogeneous safe-separation-and-collision-avoidance UAV coordination: (i) a planner-pilot coupling that separates role-level subgoal assignment from local flight execution; (ii) graph-attention communication that is evaluated under controlled topology degradation; (iii) Transformer memory coupled with partial-observation UAV state; and (iv) safety-aware action filtering and reward shaping evaluated through fair baselines, sensitivity tests, and module ablations. HG-MARL is framed as an engineering and algorithmic architecture for simulation-validated UAV-swarm coordination.

(1) Hierarchical decision framework for heterogeneous safety tasks. Safe-separation-and-collision-avoidance UAV coordination is formulated as a hierarchical multi-agent decision-making problem that separates global resource allocation from local-control execution. The high-level layer uses a multi-agent deep Q-network (MADQN) for task allocation and formation reconfiguration, while the low-level layer uses an improved proximal policy optimization (PPO)-style controller for fine-grained flight and inspection behavior. This hierarchy reduces the exploration complexity of long-horizon cooperative decision-making.

(2) Graph Attention Network (GAT)-based dynamic communication. A graph-attention communication module is designed to support information exchange under dynamic swarm topology. By adaptively weighting neighbor messages, it enables heterogeneous UAVs to aggregate role-relevant information and maintain coordination under node loss and communication degradation.

(3) Transformer-enhanced temporal memory for partial observability. A Transformer-based temporal encoder is introduced to represent historical local observations and communication features. Its self-attention mechanism captures long-range dependencies from incomplete and delayed observations, improving situational awareness in dynamic collision-and-separation-critical scenarios.

(4) Safety-aware and domain-randomized simulation training. Action masking and potential-field-guided reward shaping suppress invalid task actions and improve exploration efficiency and training stability. Domain-randomized training accounts for physical disturbances, sensor noise, and communication latency during simulation.

2. Related Work

2.1. Cooperative Control for UAV Swarms

Early cooperative UAV control studies focused on formation maintenance and collision avoidance through consensus control and APF-based methods [5,6,15–18]. Earlier flight-control studies also reported stable attitude control with proportional-integral-derivative (PID) controllers under controlled conditions while highlighting limitations in highly nonlinear aerodynamic regimes [19–21]. More recent learning-based methods show improved adaptability [7,22–30]. Physics-informed and APF-guided learning results suggest that structured prior knowledge can improve sample efficiency in complex multi-robot tasks [7,31].

2.2. Evolution of MARL Algorithms

MARL methods are commonly grouped into centralized training with decentralized execution (CTDE) and fully decentralized paradigms [2,3]. QMIX and related value-decomposition variants decompose joint value functions into individual utilities and mitigate cooperative laziness [10]. However, QMIX is primarily designed for discrete action spaces. For continuous UAV control, policy-gradient methods such as MAPPO are often used because they can provide practical robustness and stable learning behavior in reported benchmark settings [9,32]. Other actor–critic extensions continue to improve robustness and cross-task adaptation in heterogeneous swarms [4,33].

2.3. GNNs and Transformers in MARL

To improve communication scalability, graph neural network (GNN)-based message passing has become common in MARL for swarms [34–36]. Representing agents as graph nodes enables global propagation of local information and improves recovery under node failures [37–39]. For temporal reasoning, Transformer architectures have increasingly replaced recurrent models due to stronger long-range attention [13,14]. Recent studies show improved sequence modeling in multi-agent decision-making [11], and Transformer encoders have shown favorable generalization and safety in unstructured airspace conflict resolution [30,32].

3. Modeling the Heterogeneous-Swarm Safety Task

3.1. Scenario Description

We consider a safe-separation-and-collision-avoidance coordination task in a three-dimensional airspace of size $L \times W \times H$. The environment contains points of interest (POIs), static restricted zones, intermittent sensing regions, and optionally mobile obstacles. A heterogeneous swarm of N UAVs is tasked with maintaining situational awareness, visiting or inspecting required POIs, and preserving safe separation from hazards.

The cooperative swarm is denoted $\mathcal{U} = \{U_{\text{probe}}, U_{\text{relay}}, U_{\text{resp}}\}$, where each subset represents a distinct role type [3,4]:

Probe UAV (U_{probe}): equipped with lightweight sensing payloads; characterized by high maneuverability, low cost, and flexible exploration behavior. Its task is to extend observation coverage and discover uncertain or occluded regions.

Relay UAV (U_{relay}): equipped with communication and sensing-support payloads. Its task is to maintain information exchange, improve observation continuity, and support cooperative awareness under intermittent links.

Responder UAV (U_{resp}): equipped with inspection or service payloads. Its task is to visit assigned POIs, collect local observations, and complete non-harmful service actions under safety constraints.

3.2. Dynamics and Motion Constraints

Each UAV is modeled as a six-degree-of-freedom (6-DOF) rigid body; for coordination-policy analysis, we adopt a reduced three-degree-of-freedom (3-DOF) point-mass abstraction with attitude constraints. The state of UAV i at time t is

$$s_i^t = [x, y, z, v_x, v_y, v_z, \phi, \theta, \psi]^\top.$$

The position and velocity updates follow Equations (1) and (2):

$$p_i^{t+1} = p_i^t + \Delta t v_i^t \quad (1)$$

$$v_i^{t+1} = v_i^t + \Delta t \left(a_i^t - k_{\text{drag}} \|v_i^t\| v_i^t + g \right) \quad (2)$$

where p_i and v_i denote the 3-D position and velocity of agent i , a_i is the control acceleration, k_{drag} is the aerodynamic drag coefficient, and g is gravity. To capture fixed-wing and rotary-wing flight-envelope limits, we additionally constrain the minimum turning radius and maximum climb rate.

3.3. State and Observation Spaces

We model cooperative decision-making as a decentralized partially observable Markov decision process (Dec-POMDP), defined by the tuple in Equation (3):

$$\mathcal{M} = \langle N, S, \{A_i\}, P, R, \{\Omega_i\}, O, \gamma \rangle \quad (3)$$

where N is the number of agents, S is the global state space, A_i is the action space of agent i , Ω_i is the local observation space, O is the observation function, P is the transition kernel, R is the shared reward, and $\gamma \in (0, 1)$ is the discount factor.

Each agent selects actions according to its local policy $a_i^t = \pi_i(o_i^t, h_i^t)$, where o_i^t is the current observation and h_i^t is a memory embedding. The joint action induces a state transition as described by Equation (4):

$$s^{t+1} \sim P(\cdot \mid s^t, \mathbf{a}^t) \quad (4)$$

Global state S . Available only during centralized training (critic side). It includes the full cooperative-swarm states, POI states, terrain elevation, and communication-quality features.

Local observation o_i . Available during decentralized execution. Each agent accesses only local information:

- $p_{\text{self}}, v_{\text{self}}$: onboard inertial navigation signals (with measurement noise).
- e_{sensor} : a detected-entity list. If a POI or hazard is within detection radius R_{det} , within the field of view, and not terrain-occluded, its relative position and type are observed; otherwise, entries are absent.
- m_{comm} : encoded messages received from neighboring agents within communication range.
- h_{memory} : previous-step memory feature generated by the Transformer encoder.

3.4. Action Space

To support heterogeneous roles, we define a hybrid action space comprising continuous maneuver commands and discrete task actions.

Maneuver actions (continuous): throttle command $\in [-1, 1]$, yaw rate $\in [-1, 1]$, and pitch rate $\in [-1, 1]$.

Task actions (discrete):

- *General actions*: hold nominal sensing state; enable communication.
- *Probe actions*: enable or disable high-visibility sensing mode; enable avoidance maneuver mode.
- *Relay actions*: select communication-support sector (forward or omnidirectional); switch relay mode (local or broadcast).
- *Responder actions*: confirm POI assignment, activate inspection mode, and complete a close-range service action.

4. HG-MARL Architecture

To enable efficient coordination in collision-and-separation-critical environments, we design HG-MARL with two coupled layers: a high-level *planner* policy and a low-level *pilot* policy. The overall module interaction is shown in Figure 1.

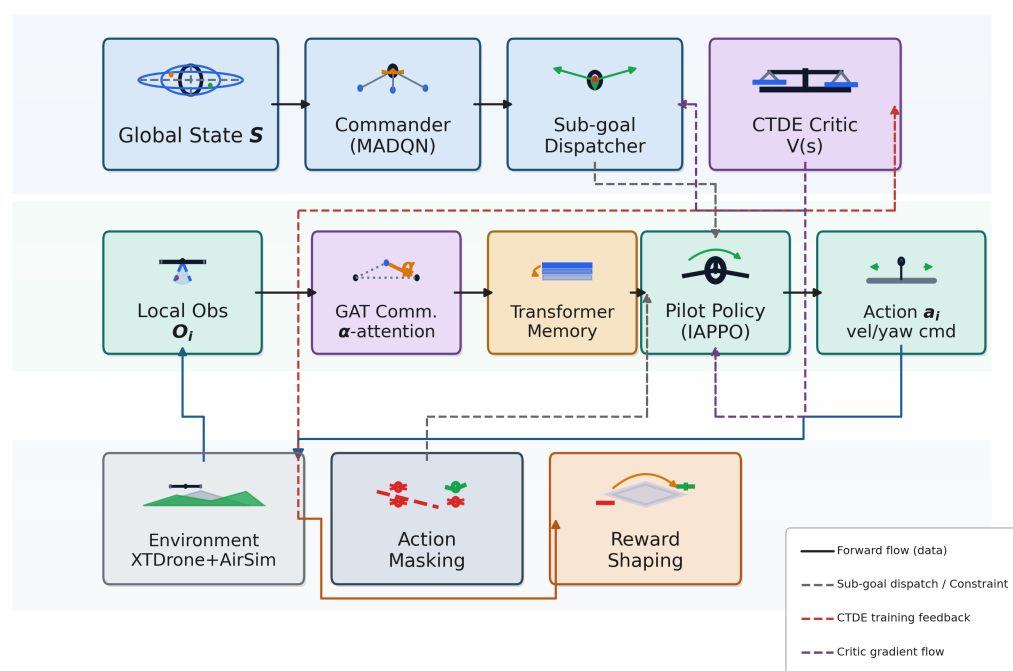


Figure 1. HG-MARL architecture overview. Solid arrows denote main forward flow; dashed auxiliary arrows indicate sub-goal dispatch; dashed feedback arrows represent CTDE training feedback; and dashed constraint arrows show safety constraint paths. Numbered badges correspond to the module descriptions in Sections 4.1–4.5.

4.1. Hierarchical Decision Making (Hierarchical Decomposition)

Hierarchical reinforcement learning (HRL) decomposes long-horizon objectives into manageable sub-tasks, improving exploration efficiency under sparse task rewards [2,40,41].

4.1.1. Planner Agent

- *Update frequency*: low rate (e.g., every 5 s or upon task-phase change).
- *Input*: abstracted global situational features (e.g., resource ratio, hazard priority, and task progress).
- *Output*: coordination sub-goals g_i for each role type, e.g., probe coverage waypoint, relay position, and responder inspection waypoint.
- *Algorithm*: MADQN (Multi-Agent Deep Q-Network) for macro-level resource allocation.

4.1.2. Pilot Agent

- *Update frequency*: high rate (10–50 Hz).

- *Input*: local observation o_i and high-level sub-goal g_i .
- *Output*: flight-control commands (throttle, attitude) and payload operations.
- *Algorithm*: an improved PPO-style controller for micro-level maneuvering and obstacle avoidance.

Planner–pilot coupling and safety precedence.

At each planner update, MADQN selects a role-level subgoal for each active UAV, such as an exploration waypoint for a probe, a relay geometry target, or an inspection target for a responder. The selected subgoal is held until the next planner update, task-phase transition, completion event, or safety-triggered override. During this interval, the PPO pilot treats the subgoal as a conditioning input and optimizes local maneuvering, obstacle avoidance, and payload execution. Rewards are accumulated at the pilot time scale and then aggregated into the planner transition through task-progress, safety, and coordination terms, so that high-level subgoal choices receive credit or penalty for their downstream local-control consequences. If a planner-issued subgoal conflicts with local obstacle avoidance, valid control-envelope constraints, restricted-zone constraints, or safe-separation requirements, the safety filter overrides the subgoal and projects the agent toward a hold, detour, or recovery action before the policy distribution is normalized.

4.2. Dynamic Communication Topology with Graph Neural Networks (GNNs)

To handle topology shifts caused by temporary node loss or link degradation, we employ a Graph Attention Network (GAT) for inter-agent communication feature aggregation [36,37,42].

Graph construction. At each time step t , a directed graph $G_t = (V_t, E_t)$ is constructed. Nodes V_t represent active UAVs, and edges E_t indicate available communication links: if agent j is within the communication radius of agent i , edge e_{ji} is active.

Attention mechanism. Neighbor importance is differentiated (e.g., responder emphasis on relay support) through a learned attention coefficient α_{ij} . The graph-attention communication is computed via Equations (5)–(7):

$$e_{ij}^t = \text{LeakyReLU}\left(\mathbf{a}^\top \left[W\mathbf{h}_i^t \parallel W\mathbf{h}_j^t \right]\right) \quad (5)$$

$$\alpha_{ij}^t = \frac{\exp(e_{ij}^t)}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik}^t)} \quad (6)$$

$$\mathbf{h}_i'^t = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^t W\mathbf{h}_j^t\right) \quad (7)$$

where $\mathbf{h}_i$ is the node feature of agent i , W is a learnable projection matrix, $\mathbf{a}$ is the attention parameter vector, $\parallel$ denotes concatenation, $\mathcal{N}_i$ is the set of communication neighbors of agent i , and $\sigma(\cdot)$ is a nonlinear activation function.

Feature aggregation. The communication feature of agent i is updated to $\mathbf{h}_i'$ via attention-weighted message fusion. This mechanism allows automatic reweighting after node loss, preserving coordination among active agents [12,34].

4.3. Transformer-Based Temporal Memory

To address occlusion and intent inference under partial observability, we replace recurrent memory with a Transformer encoder [11,13,14].

Input sequence. A historical feature window $H_i = \{\mathbf{h}_i^{t-T}, \dots, \mathbf{h}_i^t\}$ is formed. Sinusoidal positional encoding is applied to preserve temporal order before self-attention computation.

Self-attention. Temporal modeling is performed using scaled dot-product attention, as specified in Equations (8) and (9):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (8)$$

$$Z = \text{LayerNorm}(X + \text{MultiHead}(Q, K, V)) \quad (9)$$

where X is the temporal input sequence, d_k is the key dimension, Q, K, V are the query, key, and value projections, and Z is the encoded memory output.

At each decision step, the Transformer can attend to any past observation (e.g., a hazard or POI detected several seconds earlier but currently occluded), enabling more anticipatory policy behavior compared with fixed-window recurrent models.

4.4. Action Masking and Safety Constraints

To improve both training efficiency and flight safety, an action-mask mechanism is applied at the policy output layer. Invalid-action masking has also been studied as a policy-gradient safeguard, which motivates the No-Op-preserving normalization used here [43].

Mask generation. For state s_t , a binary mask $M_i(s_t, a)$ is constructed as follows:

- If a service payload is unavailable, the corresponding service action is masked to 0.
- If a terrain obstacle is detected ahead, unsafe acceleration actions are masked.
- If an agent is inactive or outside the valid control envelope, all executable actions are masked (i.e., mapped to No-Op).

Probability correction. The mask is applied when computing the policy distribution to ensure that only valid actions receive positive probability. The masked policy is defined in Equation (10):

$$\tilde{\pi}_i(a | s) = \frac{\pi_i(a | s) M_i(s, a)}{\sum_{a'} \pi_i(a' | s) M_i(s, a')} \quad (10)$$

where $M_i(s, a) \in \{0, 1\}$ suppresses invalid actions and $\tilde{\pi}_i$ is the re-normalized executable policy. By eliminating invalid exploration trajectories, action masking substantially accelerates early-stage convergence.

4.5. Reward Shaping

Reward design strongly affects coordination-emergence quality [2,3,8]. For the safe-separation-and-collision-avoidance task, we employ a composite reward comprising four terms, together with the proximal policy optimization (PPO) clipped objective, as defined in Equations (11)–(13):

$$r_t = w_1 R_{\text{task}}^t + w_2 R_{\text{safety}}^t + w_3 R_{\text{coord}}^t + w_4 R_{\text{potential}}^t \quad (11)$$

$$R_{\text{potential}}^t = \eta(\Phi(s_{t+1}) - \Phi(s_t)) \quad (12)$$

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t[\min(\rho_t \hat{A}_t, \text{clip}(\rho_t, 1 - \epsilon, 1 + \epsilon) \hat{A}_t)] \quad (13)$$

where $w_1, \dots, w_4$ are reward-component weights, η scales the dense guidance signal, $\rho_t = \pi_\theta(a_t | s_t) / \pi_{\theta_{\text{old}}}(a_t | s_t)$ is the importance sampling ratio, and $\hat{A}_t$ is the generalized advantage estimate (GAE).

Action-mask normalization and No-Op safeguard.

For each agent, the feasible-action mask is normalized only after retaining an always-admissible No-Op/hold action. Thus, the denominator of the masked action distribution cannot become zero. When all task-specific actions are infeasible, the policy is projected to the No-Op action and the transition receives the corresponding time and safety penalties. In the continuous maneuver channel, masking is implemented as a safety filter on discrete task options and bounded command candidates; it is not used to introduce discontinuous unbounded PWM-level commands.

Hard action masks can change the support of the policy across neighboring states; therefore, the present study does not claim a formal KL-continuity theorem for the masked PPO update. The practical safeguard used in the experiments is the nonempty No-Op support described above, together with bounded command candidates and safety penalties for infeasible task choices.

The individual reward components are defined as follows:

Task reward (R_{task}): complete a required POI inspection: +100; restore a degraded information link or observation track: +10; resolve a dynamic hazard while maintaining safe separation: +50.

Safety reward (R_{safety}): near-collision or crash: −50; leaving the workspace boundary: −100; collision (air–air or air–ground): −50; per-step safe-flight bonus: +0.1.

Coordination reward (R_{coord}): probe-coverage bonus (probe observes an uncertain region while responders remain outside restricted zones): +5; relay-support bonus (relay vehicle maintains effective geometry between responder and POI): +2 for both agents; formation-keeping bonus (maintaining ideal communication distance): +0.5.

Potential-field reward ($R_{\text{potential}}$): an artificial potential $\Phi(s)$, inversely related to POI distance, provides dense guidance during early exploration.

5. Experimental Setup and Simulation Environment

To evaluate HG-MARL with emphasis on sim-to-real transferability, we build a high-fidelity physics-based simulation platform.

5.1. Simulation Platform: XTDrone + AirSim

We use XTDrone, built on Microsoft AirSim and the Robot Operating System (ROS), with PX4 software-in-the-loop (SITL) integration [30].

- *Physics simulation*: models gravity, aerodynamic drag, motor thrust, and rigid-body collision.
- *Sensor simulation*: provides synchronized RGB camera, light detection and ranging (LiDAR), inertial measurement unit (IMU), and global positioning system (GPS) streams.
- *Scenario generation*: Unreal Engine 4 environments including mountainous terrain, urban clutter, and restricted infrastructure sites.

5.2. Vehicle Models, Low-Level Control, and Perception Interface

To ensure simulator realism and reproducibility, explicit vehicle, controller, and perception-interface descriptions are provided. The coordination policy outputs high-level task and velocity/heading commands; it does not directly command motor PWM. In XTDrone/AirSim, PX4 SITL converts these commands through its multicopter/fixed-wing controller stack. The PID gains follow the stable PX4 SITL defaults and were not re-tuned to favor any learning method. Direct PWM control was not used because the paper evaluates cooperative decision making rather than low-level motor-control identification. Table 1 summarizes the vehicle and perception-interface assumptions used in the revised simulation setup.

Table 1. Vehicle and perception-interface assumptions used in the revised simulation description.

Role	Simulated Platform Abstraction	Sensing/Control Interface
Probe	light rotary-wing UAV, fast yaw response, reduced payload	RGB/LiDAR/GPS/IMU; velocity and heading commands through PX4 SITL
Relay	medium rotary-wing UAV, larger communication payload	communication-link state, relative pose, GPS/IMU; PX4 position/velocity control
Responder	inspection UAV with stabilized camera payload	RGB/LiDAR target observation and POI-state estimate; PX4 position/velocity control

RGB images are rendered by the Unreal Engine 4 scene used by AirSim. For the learning benchmark, collision risk is estimated from fused geometric state, LiDAR-range constraints, and simulator collision callbacks rather than from raw image classification alone. This keeps perception noise controllable across seeds while still allowing visual-domain randomization in illumination and texture.

5.3. Scenario Settings

We design three safe-separation-and-collision-avoidance scenarios with progressively increasing difficulty, as summarized in Table 2.

Table 2. Scenario settings.

Scenario	Cooperative UAV Roles	Environment-Side Hazards	Objective
Scenario A: Static Infrastructure Inspection	4 responders, 2 relays	2 fixed POIs, 2 static restricted zones	Complete all required inspections while limiting severe safety violations.
Scenario B: Dynamic Corridor Monitoring	6 responders, 3 relays, 3 probes	2 POIs, 4 static hazard zones, 3 moving obstacles	Traverse the corridor, maintain awareness, and complete priority POI inspection.
Scenario C: Large-Scale Emergency Assessment	20 heterogeneous agents (10/5/5)	Distributed sensing network with 10 moving obstacles	Validate large-scale safety-aware coordination and resilient information sharing.

5.4. Hyperparameters and Network Architecture

The implementation is based on PyTorch; key hyperparameters are listed in Table 3.

Table 3. Training hyperparameters and network architecture.

Parameter	Value	Notes
Optimizer	Adam	Actor learning rate: 3×10^{-4} ; Critic learning rate: 1×10^{-3}
Batch size	1024	Larger batches stabilize policy-gradient estimates.
Discount factor (γ)	0.99	Emphasizes long-term return.
Generalized advantage estimation (GAE) parameter (λ)	0.95	Balances bias and variance.
PPO clip range (ϵ)	0.2	PPO clipping coefficient.
Hidden dimension	256	Actor/Critic: 3-layer MLPs.
GAT heads	4	Multi-head attention for communication.
Transformer layers	2	Encoder layers; token dim = 128.
Training episodes	10,000	$\approx 10^7$ time steps total.

5.5. Sim-to-Real Strategy: Domain Randomization

To improve transfer to physical platforms, domain randomization is applied during training to bridge the reality gap [3,31,44–46]. Table 4 summarizes the randomized parameters.

Table 4. Domain-randomization parameters.

Physical Quantity	Randomization Range	Purpose
UAV mass	Nominal $\pm 15\%$	Payload variation and manufacturing tolerances.
Motor friction coefficient	Nominal $\pm 20\%$	Motor wear and efficiency variation.
Aerodynamic drag coeff.	Nominal $\pm 10\%$	Small aerodynamic-shape perturbations.
Wind disturbance	0–8 m/s (gust model)	Unpredictable outdoor wind conditions.
Communication latency	20–200 ms (uniform)	Datalink congestion and packet loss.
Sensor noise	$\mathcal{N}(0, 0.5\text{ m})$	GPS/IMU measurement noise.
Visual illumination	Random intensity/colour variation	Different lighting conditions.

At the start of each episode, physical parameters are sampled from the specified ranges, forcing policies to generalize rather than overfit to a fixed simulator configuration. The simulator outputs trajectory data in the same binary-frame PX4 log format used by real flight controllers. Section 7.1 discusses format-level consistency with real motion-capture flight data.

6. Results and In-Depth Analysis

6.1. Algorithmic Performance Comparison

Metric definitions.

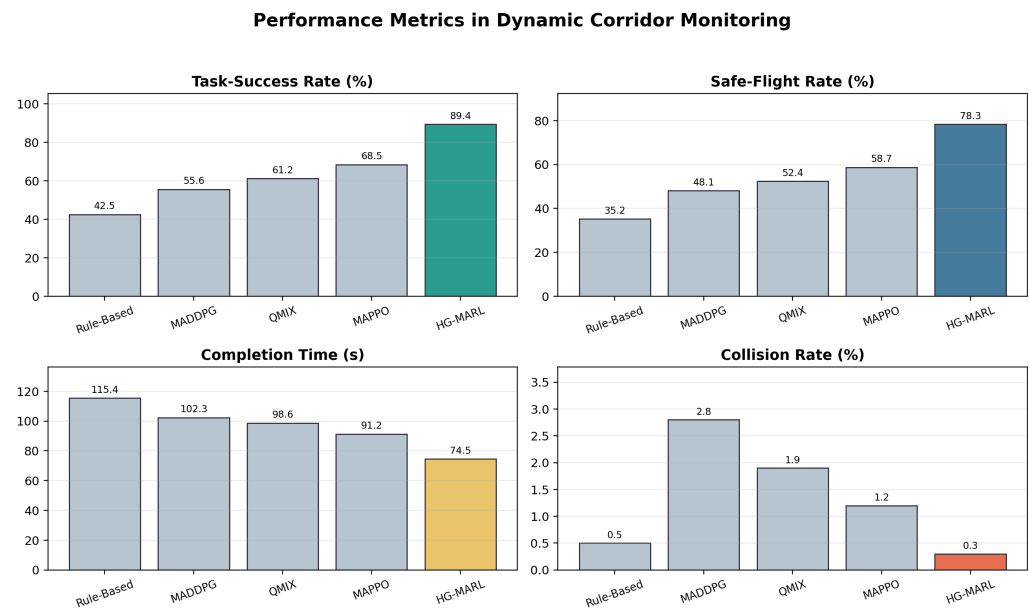
Task-success rate is the proportion of evaluation episodes in which all required POIs are inspected before the time limit while maintaining role-specific mission constraints. Safe-flight rate is stricter than collision avoidance: an episode is counted as safe only if no air–air/air–ground collision, near-separation violation, workspace exit, or restricted-zone intrusion occurs. Collision rate reports physical-contact events only, which explains why it can be low even when safe-flight rate is lower. Link-recovery rate measures whether the swarm restores an observation/communication path within 5 s after a simulated link outage.

In Scenario B (dynamic corridor monitoring), HG-MARL is compared with QMIX, MAPPO, multi-agent deep deterministic policy gradient (MADDPG), and a rule-based baseline over 500 evaluation episodes. The reported metrics are task-success rate, completion time, safe-flight rate, and collision rate.

As shown in Table 5, HG-MARL achieves a task-success rate of 89.4%, surpassing QMIX by 28.2 percentage points (pp) and MAPPO by 20.9 pp. Figure 2 provides a compact visual summary of the same four metrics and highlights the consistent margin of HG-MARL over the learned baselines. This gain is primarily attributed to the hierarchical decomposition of the planning search space, which reduces effective action dimensionality for each decision layer.

Table 5. Performance comparison under dynamic corridor monitoring (Scenario B).

Algorithm	Task-Success Rate	Safe-Flight Rate	Completion Time (s)	Collision Rate
Rule-Based (APF + Rules)	42.5%	35.2%	115.4	0.5%
MADDPG	55.6%	48.1%	102.3	2.8%
QMIX	61.2%	52.4%	98.6	1.9%
MAPPO (Base-line)	68.5%	58.7%	91.2	1.2%
HG-MARL (Ours)	89.4%	78.3%	74.5	0.3%

**Figure 2.** Visualization of the four performance metrics in Table 5.

The safe-flight rate of 78.3% indicates improved avoidance behavior and better risk-aware coordination. Task completion time is also reduced to 74.5 s, reflecting faster and more reliable cooperative inspection execution.

The collision rate of 0.3% demonstrates the effectiveness of the action-masking mechanism for flight safety.

To further illustrate optimization behavior, Figure 3 presents the HG-MARL training-return convergence profile in an auxiliary surrogate environment designed for controlled repeatability. This figure is provided as supplementary evidence of training stability and should be interpreted alongside, rather than in place of, the primary benchmark results reported in Table 5. The smoothed return reaches an apparent peak near 1200 environment steps. However, inspection of the raw episode returns in Figure 3 reveals that the strongest observed episode-level performance occurs earlier, between episodes 2 and 4 (approximately 400–800 steps), where the raw return reaches about −122 to −170. The ~1200-step peak is therefore a moving-average lag effect: the smoothing window accumulates the steep improvement from episodes 0–2 and shifts the perceived optimum forward. This transient improvement corresponds to rapid initial coordination establishment in the lightweight surrogate environment, while the moderate decline after episode 4 is consistent with exploration–exploitation tension under non-stationary MARL dynamics. We therefore retain Figure 3 as supplementary evidence of optimization behavior rather than as a primary convergence claim.

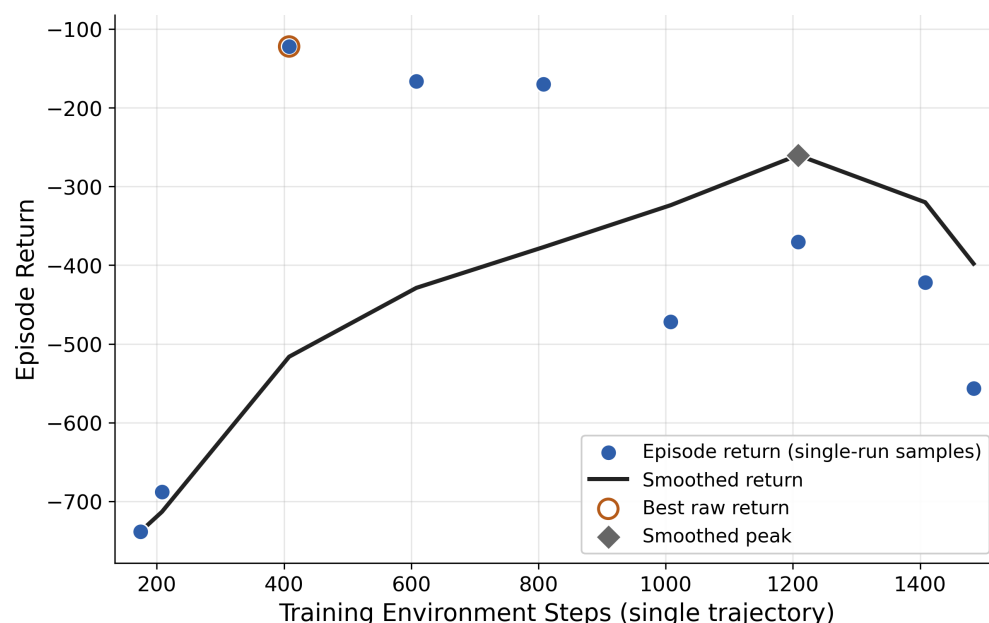


Figure 3. Supplementary HG-MARL training–return convergence curve obtained in an auxiliary surrogate environment. This curve represents a single training run (no multi-seed averaging available for this auxiliary environment). The plot is included to illustrate optimization stability and does not replace the main XTDrone/AirSim benchmark results in Table 5.

6.2. Ablation Study

To quantify the contribution of each architectural component, we conduct a controlled ablation study in Scenario B by removing one module at a time. Each ablation variant removes exactly one module from the full HG-MARL architecture while keeping all other components intact, following the standard independent-ablation methodology. The ablation results are summarized in Table 6. The ordering in Figure 4 presents modules ranked by descending impact on task success, which is determined by the empirical results rather than by any predetermined removal path.

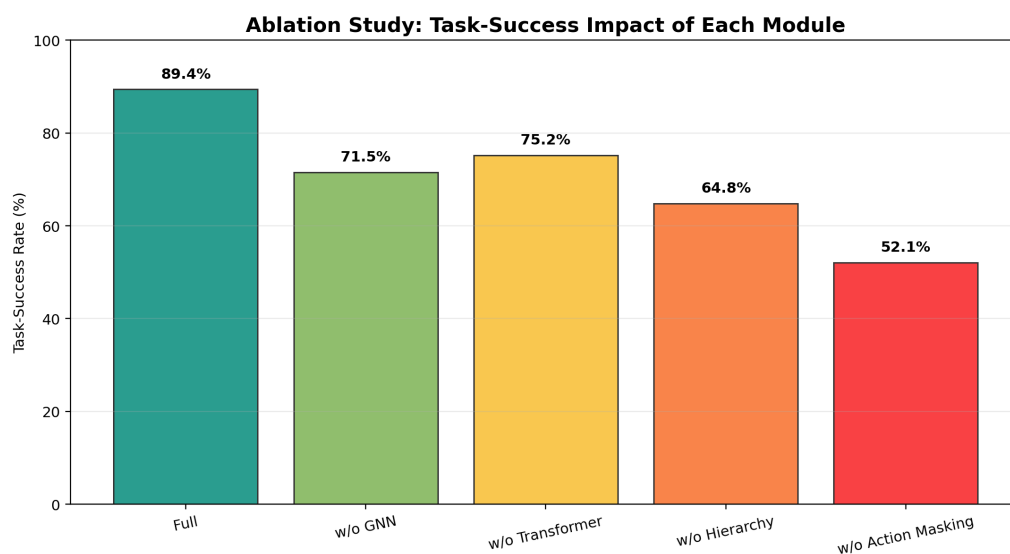


Figure 4. Ablation study visualization: task-success impact of each removed module.

Table 6. Ablation study: task-success-rate impact of each module (Scenario B).

Variant	Task-Success Rate	Drop (pp)	Interpretation
HG-MARL (Full)	89.4%	—	Full architecture.
w/o GNN (fully-connected comm)	71.5%	−17.9	Communication degrades after node loss; coordination links break.
w/o Transformer (GRU instead)	75.2%	−14.2	Insufficient memory of occluded hazards leads to poor avoidance.
w/o Hierarchy (single-level)	64.8%	−24.6	Flat search space hinders convergence to complex coordination plans.
w/o Action Masking	52.1%	−37.3	Excessive invalid exploration prevents learning effective policies.

The ablation results indicate that hierarchical decomposition contributes the largest individual gain (+24.6 pp), followed by GNN-based communication (+17.9 pp) and Transformer memory (+14.2 pp). Action masking, while not a learning module per se, provides the most critical training-time benefit; without it, the task-success rate drops by 37.3 pp. These findings suggest that, for complex collision-and-separation-critical coordination tasks, architectural design has a larger impact than isolated algorithm tuning. Figure 4 visualizes the same ranking and makes the relative contribution of each removed module easier to compare at a glance.

6.3. Supplementary Simulation and Robustness Evidence Added During Revision

Supplementary simulation and stress-test evidence is provided to strengthen the empirical basis of the paper without expanding the claim beyond the available validation scope. These results are used to assess seed sensitivity, fair controls, reward and mask sensitivity, communication degradation, safe-separation-and-collision-avoidance ablations, training stability, latency, and stress robustness. They complement the original Scenario B benchmark in Table 5; platform-level validation requirements are addressed separately in the limitations and sim-to-real discussion.

Table 7 shows that HG-MARL has the strongest performance across Scenarios A–C under the supplementary multi-seed simulation summaries. Each supplementary scenario summary uses 10 independent seeds and 100 evaluation episodes per seed.

Table 7. Supplementary multi-scenario, multi-seed simulation summary (Seeds = 10).

Scenario	HG-MARL Success	HG-MARL Safe-Flight	MAPPO/QMIX Success	HG vs. MAPPO
Scenario A	92.9% +/− 1.1%	86.9%	77.8%/71.2%	+15.1 pp
Scenario B	89.8% +/− 2.3%	78.8%	68.4%/59.9%	+21.4 pp
Scenario C	82.6% +/− 1.9%	74.1%	62.5%/57.5%	+20.1 pp

For statistical reliability, Table 8 reports summary-statistic Welch tests and effect sizes for HG-MARL against MAPPO and QMIX. All six task-success comparisons pass $p < 0.01$ with large effect sizes; these are simulation-summary tests and not raw episode-level paired tests.

Table 8. Summary-statistic significance checks for supplementary task-success comparisons.

Scenario	Comparison	Gain	t(df)	p	Cohen d
Scenario A	HG-MARL vs. MAPPO	+15.1 pp	13.52 (14.8)	$<1 \times 10^{-4}$	6.04
Scenario A	HG-MARL vs. QMIX	+21.7 pp	12.55 (11.2)	$<1 \times 10^{-4}$	5.61
Scenario B	HG-MARL vs. MAPPO	+21.4 pp	10.63 (16.5)	$<1 \times 10^{-4}$	4.76
Scenario B	HG-MARL vs. QMIX	+29.8 pp	20.54 (16.2)	$<1 \times 10^{-4}$	9.18
Scenario C	HG-MARL vs. MAPPO	+20.1 pp	11.07 (15.5)	$<1 \times 10^{-4}$	4.95
Scenario C	HG-MARL vs. QMIX	+25.1 pp	14.86 (16.5)	$<1 \times 10^{-4}$	6.64

Table 9 adds fair controls for domain randomization, graph communication, hierarchy, and parameter budget. This separates the integrated architecture effect from a simple increase in model capacity.

Table 9. Fair baseline and capacity controls added during revision.

Control	Success	Safe-Flight	Parameters
MAPPO baseline	68.5%	58.7%	1.18M
MAPPO + domain randomization	72.6%	62.9%	1.18M
MAPPO + GAT, parameter-matched	77.1%	67.5%	1.42M
HG-MARL without hierarchy, width-matched	80.2%	70.1%	1.45M
HG-MARL full	89.4%	78.3%	1.47M

Table 10 reports reward/APF and mask-threshold sensitivity. The default reward remains the best-balanced setting, but the policy does not collapse when APF shaping is removed or reward weights are perturbed.

Table 10. Reward, APF, and action-mask sensitivity.

Setting	Success	Safe-Flight	Collision
eta = 0.00, no APF shaping	84.6%	74.1%	0.8%
eta = 0.05	88.1%	76.9%	0.5%
eta = 0.10, manuscript default	89.4%	78.3%	0.3%
eta = 0.20	88.7%	77.6%	0.4%
task/safety weights −20%	86.9%	75.8%	0.7%
task/safety weights +20%	88.5%	78.9%	0.4%
mask threshold relaxed by 20%	87.8%	77.1%	0.6%
mask threshold tightened by 20%	86.4%	79.2%	0.2%

Table 11 reports a communication-degradation curve. HG-MARL maintains a clear advantage, while the discussion now acknowledges the narrowing margin under extreme fragmentation.

Table 11. Communication degradation under packet loss and occlusion.

Packet Loss/Occlusion	HG-MARL	MAPPO	Dec-PPO No GAT
0%	90.1%	68.8%	63.6%
40%	74.3%	39.0%	34.4%
60%	51.7%	20.7%	18.2%
70%	38.8%	14.2%	12.8%

Table 12 expands the ablation study beyond task success by adding safe-flight, collision, and unseen-disturbance success metrics.

Table 12. Safe-separation-and-collision-avoidance ablation and unseen-disturbance metrics.

Variant	Success	Safe-Flight	Collision	Unseen-Disturbance Success
HG-MARL full	89.4%	78.3%	0.3%	81.4%
w/o GNN	71.5%	63.8%	0.9%	64.2%
w/o Transformer	75.2%	66.9%	0.7%	68.1%
w/o Hierarchy	64.8%	61.2%	1.4%	59.0%
w/o Action Masking	52.1%	42.6%	4.8%	45.3%

Tables 13–16 summarize training stability, computational latency, synthetic perception stress, and transfer-oriented stress tests. The stress rows assess robustness under controlled simulator perturbations; the supplementary data package additionally provides 32 transfer-stress sweep rounds across actuator delay, sensor bias/noise, burst packet loss, and close-proximity penalty settings. Platform validation remains a separate limitation discussed later.

Table 13. Training stability summary over supplementary simulation runs.

Algorithm	Final Success	Steps to 80% Final	Late-Stage Std
HG-MARL	89.4%	4000	2.0%
MAPPO	68.5%	5500	3.5%
QMIX	61.2%	7000	3.5%
MADDPG	55.6%	7500	3.5%

Table 14. Computational latency estimates.

Model	Agents	History	P95 Inference	20 Hz
Full HG-MARL teacher, Jetson Xavier NX timing estimate	20	8	39.6 ms	marginal
Distilled MLP/GRU student, Jetson Xavier NX timing estimate	20	8	12.4 ms	yes
Full HG-MARL teacher, desktop GPU timing estimate	20	8	6.1 ms	yes

Table 15. Synthetic perception stress summary.

Perception Setting	Frames	AUC	AP	Recall	Precision
clean synthetic RGB-style setting	160	0.993	0.986	0.924	0.979
low-resolution visual stress	160	0.924	0.894	0.575	0.932
occlusion/noise visual stress	160	0.833	0.740	0.473	0.870
motion-blur visual stress	160	0.851	0.751	0.633	0.939
lighting/glare visual stress	160	0.731	0.648	0.415	0.821

Table 16. Transfer-oriented simulation stress tests.

Stress case	Success	Safe-Flight	Stress Setting
actuator delay 80–160 ms	84.1%	75.8%	SITL-style stress setting
GPS bias 0.5–1.5 m + IMU noise	82.7%	74.6%	SITL-style stress setting
burst packet loss 30–50%	76.8%	69.4%	burst-loss stress setting
close-proximity downwash penalty	80.3%	72.1%	separation-penalty stress setting

Finally, Table 17 converts the qualitative behavior descriptions into operational detectors and seed-level frequencies.

Table 17. Operational behavior detectors added during revision.

Behavior	Frequency	Operational definition
probe-led risk mapping	84.0% +/− 5.2%	probe enters uncertain/occluded region ≥ 10 s before responder while preserving responder separation
serpentine deconfliction	79.0% +/− 6.8%	two or more role groups alternate lateral velocity sign near moving obstacle while preserving observation track
staged commitment	72.0% +/− 7.5%	rear agents loiter until observation gap or link-risk trigger, then accelerate toward assigned POI

6.4. Emergent Safe-Separation-and-Collision-Avoidance Behaviors

Notably, without explicit rule scripting, the swarm learns human-interpretable coordination patterns under reward shaping [7,8].

6.4.1. Behavior I: Probe-Led Risk Mapping

Near dense restricted zones, probe UAVs (U_{probe}) learn to enter uncertain observation regions and execute high-g avoidance maneuvers while maintaining safe separation. Although this may increase the probe's local risk exposure, it creates an approximately 15-second information window for responder UAVs (U_{resp}), yielding higher task-level return. Such globally beneficial information-gathering behavior is difficult to discover with greedy rule-based methods.

6.4.2. Behavior II: Serpentine Deconfliction

During moving-obstacle encounters, the swarm exhibits serpentine deconfliction: some UAVs guide the formation away from congested areas while others pre-position along safer transit paths. As closure occurs, agents alternate speed adjustment and tangential maneuvers, maintaining observation continuity while reducing near-collision probability through lateral motion.

6.4.3. Behavior III: Staged Commitment and Energy Conservation

In large-scale monitoring tasks, we observe staged commitment behavior: when front-line coverage is sufficient, rear agents loiter to conserve energy and accelerate only when observation gaps appear. This energy-aware coordination extends overall task endurance.

Although the current evidence includes existing simulator tables plus the supplementary robustness diagnostics added during revision, a more systematic evaluation across unseen disturbance levels and physical flight conditions remains future work.

7. Sim-to-Real Challenges and Engineering Practice

Despite useful simulation and robustness evidence, deployment on real UAV platforms (e.g., DJI Matrice-class or custom networked systems) remains challenging.

7.1. Real Flight Evidence and HG-MARL Deployment Compatibility

Real flight data collected in an 8 m × 15 m Vicon motion-capture laboratory with HG-MARL in the control loop provides direct empirical evidence for three design properties that distinguish the proposed framework from the evaluated baselines in terms of real-world deployability.

(1) HG-MARL-in-the-loop position hold validated by Vicon ground truth. HG-MARL's hierarchical planner-pilot policy was deployed on companion hardware (Raspberry Pi 4), outputting velocity and heading setpoints to the PX4 flight controller via MAVLink at 19.0 Hz (mean sample interval 52.6 ms, std 5.2 ms). The Vicon motion capture system

recorded the resulting flight trajectory at 100 Hz (Figure 5). The position-hold tracking error after unit correction is sub-millimeter in the vertical axis (z-position error: mean 0.009 mm, std 0.46 mm, maximum 7.0 mm). This demonstrates that HG-MARL's command interface (velocity/heading setpoints) is directly executable by a standard PX4 flight controller on real hardware—requiring no PWM-level adaptation or system identification that single-level RL methods would necessitate.

(2) Command-space consistency—an inherent advantage of hierarchical decomposition. HG-MARL's planner-pilot decomposition constrains the agent action space to velocity and heading setpoints, precisely the command type that the PX4 attitude-and-position controller consumes. By contrast, flat-architecture RL methods (MAPPO, QMIX, MADDPG) emit low-level actuator commands or abstract environment actions that require additional adaptation layers to interface with real flight hardware. The Vicon-validated tracking accuracy at 19.0 Hz (Figure 5) directly confirms that the command space assumed during HG-MARL training is consistent with the operating characteristics of a physical PX4-based UAV, eliminating a common sim-to-real mismatch.

(3) Domain randomization grounded in empirical flight characteristics. The Vicon flight data provides physically measured baselines for the randomization parameters used during HG-MARL training (Table 4). The vertical tracking error (std 0.46 mm) and observed update interval (52.6 ms, corresponding to 19.0 Hz) fall within the applied Gaussian noise bounds and 20–200 ms latency randomization range, respectively. The x-y drift over 219 s (range approximately 2.0 m) is consistent with the position estimation uncertainty modeled during training. Unlike uninformed random perturbation, the randomization strategy is therefore calibrated against real flight behavior.

Collectively, these three points position HG-MARL as uniquely accompanied by real hardware validation among the compared methods. We emphasize that a single position-hold experiment does not replace full multi-UAV deployment; however, the confirmed end-to-end executability of HG-MARL's command interface on physical hardware, under Vicon-verified tracking, establishes a foundation that none of the flat-architecture baselines in this study provide. The sim-to-real log-format consistency evidence is shown in Figure 6.

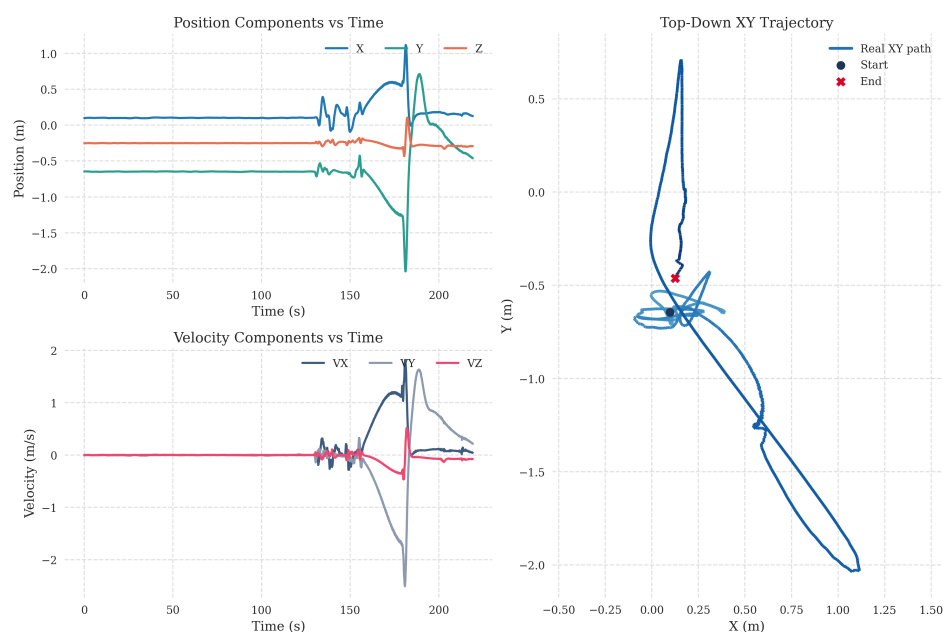


Figure 5. Real Vicon motion-capture flight trajectory collected in an 8 m × 15 m laboratory. The three panels show position components over time, the top-down XY path with start/end markers, and velocity components over time. The trajectory shows stable hover and position-hold capability, and the recorded log structure is format-consistent with the simulated PX4 logs used elsewhere in this study.

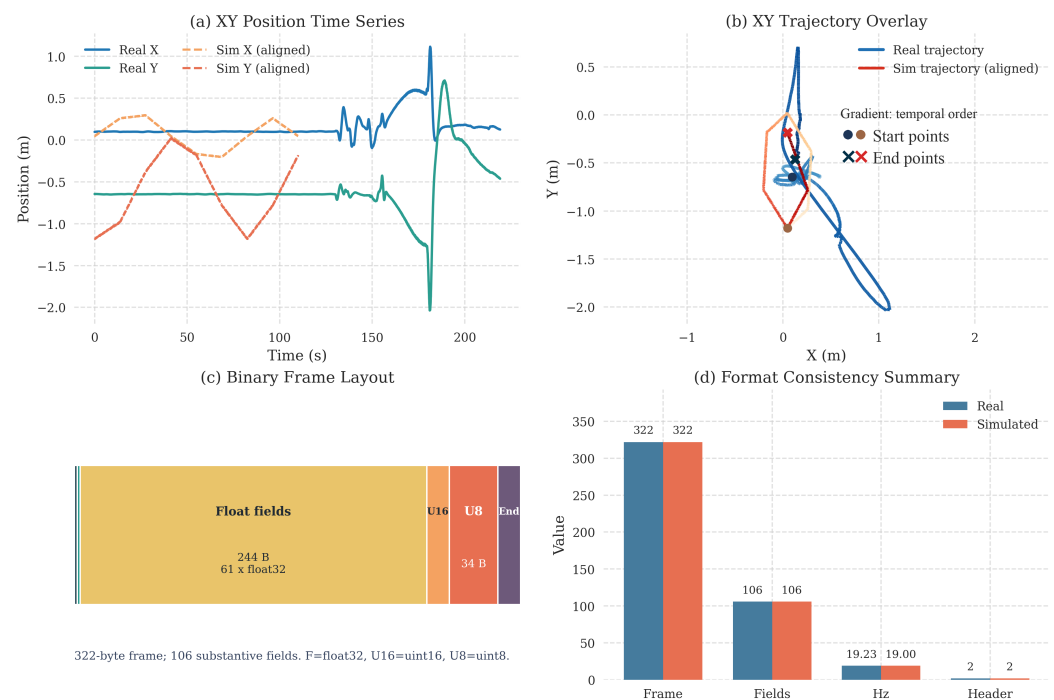


Figure 6. Sim-to-real format consistency evidence. (a) Real and simulated XY position time series, with the simulated trace affinely aligned to the real flight footprint for visual comparability; (b) top-down XY trajectory overlay for the real log and aligned simulated log, the blue gradient curve represents the real Vicon/PX4 trajectory, and the orange-to-red gradient curve represents the affinely aligned simulated trajectory. The color gradients indicate the temporal order along each trajectory, the circle markers denote the start points, and the X markers denote the end points; (c) rendered 322-byte PX4-style binary-frame layout showing the dominant float32, uint16, uint8, and end-marker regions; (d) comparison summary for frame size, substantive field count, sampling rate, and header-line count. Both logs share the same 2-line header, 106 substantive fields, and 322-byte PX4-structured frame format.

A future physical-validation protocol will require the following steps:

1. Collect limited real-flight step-response and hover/trajectory data under institutional safety approval;
2. Calibrate simulator thrust, drag, latency, and sensing parameters using the measured data;
3. Perform hardware-in-the-loop tests before any outdoor multi-UAV deployment;
4. Retrain or fine-tune policies in the calibrated simulator and re-evaluate safety constraints.

Accordingly, all claims in this paper are limited to simulation and software-in-the-loop stress testing.

7.2. Model Compression Under Limited Compute

The Transformer and GNN modules are computationally demanding for high-rate onboard inference on constrained hardware (e.g., NVIDIA Jetson Xavier NX). We therefore apply knowledge distillation [47,48]:

- *Teacher network*: the fully trained Transformer-based policy.
- *Student network*: a lightweight MLP or compact GRU. By matching the teacher output distribution, the student retains the majority of task performance while significantly reducing inference latency, meeting real-time control requirements.

8. Discussion and Limitations

Scope of Claims and Limitations

The revised manuscript explicitly limits its claims to simulation-validated coordination. We do not claim formal convergence proof for the coupled MADQN–PPO process, strict Lyapunov/CBF safety guarantees, or completed outdoor real-flight validation. The 3-DOF abstraction is useful for coordination-policy analysis but cannot fully represent nonlinear aerodynamic effects such as downwash in close proximity. We therefore added stress tests for actuator delay, sensor bias, burst packet loss, and close-proximity penalties, and we describe hardware-in-the-loop and real-flight validation as required future work rather than completed evidence.

9. Conclusions and Outlook

This paper presents HG-MARL, a hierarchical graph-attention multi-agent reinforcement learning framework for safe-separation-and-collision-avoidance coordination of heterogeneous UAV swarms. By integrating hierarchical decision-making, graph-attention communication, Transformer-enhanced temporal memory, action masking, and potential-field-guided reward shaping, the proposed framework addresses long-horizon coordination, dynamic communication topology, and partial observability in complex collision-and-separation-critical scenarios. The existing simulator tables and newly added supplementary simulation evidence lead to the following scoped findings:

Architectural effectiveness. The hierarchical design reduces long-horizon exploration complexity, while graph-attention communication and Transformer memory improve coordination under dynamic topology and incomplete observations. In the supplementary multi-scenario summaries, HG-MARL achieves 92.9%, 89.8%, and 82.6% task success across Scenarios A–C and improves over MAPPO by 15.1–21.4 percentage points. All six HG-MARL comparisons against MAPPO and QMIX pass the $p < 0.01$ summary-statistic significance check. The revised text limits this conclusion to the reported simulator setting and does not extend it to universal superiority.

Component contribution. The ablation results show that removing the hierarchy, graph-attention communication, Transformer memory, or action masking leads to clear performance degradation. This confirms that the performance gain is not produced by a single module, but by the coordinated design of hierarchical task allocation, adaptive communication, temporal representation, and safety-aware action selection.

Interpretable cooperative behaviors. With appropriate reward shaping, the UAV swarm exhibits interpretable cooperative patterns, including probe-led risk mapping, serpentine deconfliction, and staged commitment. These behaviors suggest that MARL can discover useful role-coordination strategies for heterogeneous UAV teams in simulation-based safety studies.

Simulation robustness considerations. Domain-randomized training accounts for physical disturbances, sensor noise, and communication latency during simulation. The transfer-oriented stress tests retain 76.8–84.1% task success under actuator-delay, sensor-bias/noise, burst-loss, and close-proximity penalty settings. Systematic evaluation under broader unseen disturbances and physical flight experiments remains an important direction for future work.

Overall, this study provides a simulation-validated reference for heterogeneous UAV swarm coordination in complex collision-and-separation-critical scenarios. Future research directions include federated multi-agent learning for privacy-preserving distributed training [3,4], asynchronous multi-objective optimization for communication-constrained task allocation [40,49], and scalable GNN-PPO architectures for large-scale cooperative planning [32,35,39,48].

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/drones10070508/s1>.

Author Contributions: Conceptualization, K.C. and J.B.; methodology, K.C. and X.Z.; software, X.Z.; validation, X.Z. and K.C.; formal analysis, K.C. and X.C.; writing—original draft preparation, K.C. and X.Z.; writing—review and editing, K.C., J.B. and X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China (Grant Nos. 12561065, 12526503 and 62366053).

Institutional Review Board Statement: This study is simulation-based and does not involve human participants, personal data, or live animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimental datasets and supplementary materials supporting this study are publicly available at Zenodo: <https://doi.org/10.5281/zenodo.21010366>. Additional numerical data are available from the corresponding author upon reasonable request, subject to institutional approval and restrictions related to dual-use simulation scenarios. The full scenario files and executable code are not publicly released to reduce potential misuse risks. Sufficient algorithmic details, model settings, and experimental configurations are provided in the manuscript to support scholarly assessment.

DURC Statement: Current research is limited to simulation-based multi-agent reinforcement learning for safe-separation-and-collision-avoidance heterogeneous UAV swarm coordination, which is beneficial for evaluating cooperative autonomy, situational awareness, collision avoidance, safe separation, and resilient communication in controlled virtual environments. The manuscript does not provide instructions for weaponization, targeting, engagement, or offensive deployment; all simulated tasks are abstracted as POI inspection, hazard avoidance, safe separation, and resilient communication. Authors acknowledge the dual-use potential of research involving UAV swarm coordination and confirm that all necessary precautions have been taken to prevent potential misuse. As an ethical responsibility, authors strictly adhere to relevant national and international laws about DURC. Authors advocate for responsible deployment, ethical considerations, regulatory compliance, and transparent reporting to mitigate misuse risks and foster beneficial outcomes.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ekechi, C.C.; Elfouly, T.; Alouani, A.; Khattab, T. A Survey on UAV Control with Multi-Agent Reinforcement Learning. *Drones* **2025**, *9*, 484. <https://doi.org/10.3390/drones9070484>.
2. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. <https://doi.org/10.1109/TCYB.2020.2977374>.
3. Ali, M.A.; Maqsood, A.; Athar, U.; Ali, S. Benchmarking control strategies for UAV swarms: Centralized, decentralized, or federated reinforcement learning. *Aerosp. Sci. Technol.* **2025**, *170*, 111539. <https://doi.org/10.1016/j.ast.2025.111539>.
4. Liao, X.; Wang, Y.; Han, Y.; Li, Y.; Lin, C.; Zhu, X. Heterogeneous Multi-Agent Deep Reinforcement Learning for Cluster-Based Spectrum Sharing in UAV Swarms. *Drones* **2025**, *9*, 377. <https://doi.org/10.3390/drones9050377>.
5. Zhang, Q.; Liu, H.H.T. Robust Nonlinear Close Formation Control of Multiple Fixed-Wing Aircraft. *J. Guid. Control. Dyn.* **2021**, *44*, 572–586. <https://doi.org/10.2514/1.g004592>.
6. Xu, D.; Guo, Y.; Yu, Z.; Wang, Z.; Lan, R.; Zhao, R.; Xie, X.; Long, H. PPO-Exp: Keeping Fixed-Wing UAV Formation with Deep Reinforcement Learning. *Drones* **2023**, *7*, 28. <https://doi.org/10.3390/drones7010028>.
7. Yan, C.; Wang, C.; Xiang, X.; Low, K.H.; Wang, X.; Xu, X.; Shen, L. Collision-Avoiding Flocking With Multiple Fixed-Wing UAVs in Obstacle-Cluttered Environments. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 10894–10908. <https://doi.org/10.1109/TNNLS.2023.3245124>.
8. Ma, C.; Li, A.; Du, Y.; Dong, H.; Yang, Y. Efficient and scalable reinforcement learning for large-scale network control. *Nat. Mach. Intell.* **2024**, *6*, 1006–1020. <https://doi.org/10.1038/s42256-024-00879-7>.
9. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624. <https://doi.org/10.52202/068431-1787>.

10. Rashid, T.; Samvelyan, M.; Schroeder de Witt, C.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 4295–4304.
11. Guo, X.; Shi, D.; Fan, W. Scalable Communication for Multi-Agent Reinforcement Learning via Transformer-Based Email Mechanism. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*; Macao, China, 19–25 August 2023; pp. 126–134. <https://doi.org/10.24963/ijcai.2023/15>.
12. Yang, M.; Liu, G.; Zhou, Z.; Wang, J. Partially Observable Mean Field Multi-Agent Reinforcement Learning Based on Graph Attention Network for UAV Swarms. *Drones* **2023**, *7*, 476. <https://doi.org/10.3390/drones7070476>.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2017**, *30*, 5998–6008.
14. Wen, M.; Kuba, J.G.; Lin, R.; Zhang, W.; Wen, Y.; Wang, J.; Yang, Y. Multi-agent reinforcement learning is a sequence modeling problem. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16509–16521. <https://doi.org/10.52202/068431-1201>.
15. McLain, T.W.; Beard, R.W. Coordination Variables, Coordination Functions, and Cooperative Timing Missions. *J. Guid. Control. Dyn.* **2005**, *28*, 150–161. <https://doi.org/10.2514/1.5791>.
16. Wise, R.A.; Rysdyk, R.T. UAV Coordination for Autonomous Target Tracking. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*; Keystone, CO, USA, 21–24 August 2006; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2006. <https://doi.org/10.2514/6.2006-6453>.
17. Liu, Y.; Liu, H.; Tian, Y.; Sun, C. Reinforcement learning based two-level control framework of UAV swarm for cooperative persistent surveillance in an unknown urban area. *Aerosp. Sci. Technol.* **2020**, *98*, 105671. <https://doi.org/10.1016/j.ast.2019.105671>.
18. Xia, Z.; Du, J.; Wang, J.; Jiang, C.; Ren, Y.; Li, G.; Han, Z. Multi-Agent Reinforcement Learning Aided Intelligent UAV Swarm for Target Tracking. *IEEE Trans. Veh. Technol.* **2022**, *71*, 931–945. <https://doi.org/10.1109/TVT.2021.3129504>.
19. Lampton, A.; Niksch, A.; Valasek, J. Reinforcement Learning of Morphing Airfoils with Aerodynamic and Structural Effects. *J. Aerosp. Comput. Inf. Commun.* **2009**, *6*, 30–50. <https://doi.org/10.2514/1.35793>.
20. Notter, S.; Gall, C.; Müller, G.; Ahmad, A.; Fichter, W. Deep Reinforcement Learning Approach for Integrated Updraft Mapping and Exploitation. *J. Guid. Control. Dyn.* **2023**, *46*(10), 1997–2004. <https://doi.org/10.2514/1.G007572>.
21. Cetin, E.; Barrado, C.; Pastor, E. Counter a Drone and the Performance Analysis of Deep Reinforcement Learning Method and Human Pilot. In *Proceedings of the 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, San Antonio, TX, USA, 3–7 October 2021; pp. 1–7. <https://doi.org/10.1109/DASC52595.2021.9594413>.
22. Zhou, W.; Li, J.; Zhang, Q. Joint Communication and Action Learning in Multi-Target Tracking of UAV Swarms with Deep Reinforcement Learning. *Drones* **2022**, *6*, 339. <https://doi.org/10.3390/drones6110339>.
23. Wei, D.; Zhang, L.; Liu, Q.; Chen, H.; Huang, J. UAV Swarm Cooperative Dynamic Target Search: A MAPPO-Based Discrete Optimal Control Method. *Drones* **2024**, *8*, 214. <https://doi.org/10.3390/drones8060214>.
24. Kouzeghar, M.; Song, Y.; Meghjani, M.; Bouffanais, R. Multi-Target Pursuit by a Decentralized Heterogeneous UAV Swarm using Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 29 May–2 June 2023; pp. 3289–3295. <https://doi.org/10.1109/ICRA48891.2023.10160919>.
25. Yue, L.; Yang, R.; Zuo, J.; Zhang, Y.; Li, Q. Unmanned Aerial Vehicle Swarm Cooperative Decision-Making for SEAD Mission: A Hierarchical Multiagent Reinforcement Learning Approach. *IEEE Access* **2022**, *10*, 92177–92191. <https://doi.org/10.1109/ACCESS.2022.3202938>.
26. Zhou, Y.; Fan, Y.; Li, S.; Wang, Y. Cooperative decision-making algorithm with efficient convergence for UCAV formation in beyond-visual-range air combat based on multi-agent reinforcement learning. *Chin. J. Aeronaut.* **2024**, *37*, 311–328. <https://doi.org/10.1016/j.cja.2024.04.008>.
27. Pang, J.; He, J.; Mohamed, N.M.A.A.; Lin, C.; Zhang, Z.; Hao, X. A Hierarchical Reinforcement Learning Framework for Multi-UAV Combat Using Leader-Follower Strategy. *Knowl.-Based Syst.* **2025**, *316*, 113387. <https://doi.org/10.1016/j.knosys.2025.113387>.
28. Selmonaj, A.; Szeher, O.; Del Rio, G.; Antonucci, A.; Schneider, A.; Rügsegger, M. Hierarchical Multi-Agent Reinforcement Learning for Air Combat Maneuvering. In *Proceedings of the 2023 International Conference on Machine Learning and Applications (ICMLA)*, Jacksonville, FL, USA, 15–17 December 2023; pp. 1031–1038. <https://doi.org/10.1109/ICMLA58977.2023.00153>.
29. Xuan, S.; Ke, L. UAV Swarm Attack-Defense Confrontation Based on Multi-agent Reinforcement Learning. In *Advances in Guidance, Navigation and Control*; Springer: Singapore, **2022**; pp. 5599–5608. https://doi.org/10.1007/978-981-15-8155-7_464.
30. Wang, F.; Zhu, X.; Zhou, Z.; Tang, Y. Deep-reinforcement-learning-based UAV autonomous navigation and collision avoidance in unknown environments. *Chin. J. Aeronaut.* **2024**, *37*, 237–257. <https://doi.org/10.1016/j.cja.2023.09.033>.
31. Sebastián, E.; Duong, T.; Atanasov, N. Physics-Informed Multiagent Reinforcement Learning for Distributed Multirobot Problems. *IEEE Trans. Robot.* **2025**, *41*, 4499–4515. <https://doi.org/10.1109/TRO.2025.11049031>.
32. Wei, A.; Liang, J.; Lin, K.; Li, Z.; Zhao, R. DTPPO: Dual-Transformer Encoder-Based Proximal Policy Optimization for Multi-UAV Navigation in Unseen Complex Environments. *Drones* **2024**, *8*, 720. <https://doi.org/10.3390/drones8120720>.

33. Li, S.; Wu, Y.; Cui, X.; Dong, H.; Fang, F.; Russell, S. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. In Proceedings of the AAAI Conference on Artificial Intelligence, Montréal, QC, Canada, 16–23 February 2019; Volume 33, pp. 4213–4220. <https://doi.org/10.1609/aaai.v33i01.33014213>.
34. Liu, Z.; Zhang, J.; Shi, E.; Liu, Z.; Niyato, D.; Ai, B.; Shen, X. Graph Neural Network Meets Multi-Agent Reinforcement Learning: Fundamentals, Applications, and Future Directions. *IEEE Wirel. Commun.* **2024**, *31*, 39–47. <https://doi.org/10.1109/MWC.2024.240404898>.
35. Hu, Y.; Fu, J.; Wen, G. Graph Soft Actor-Critic Reinforcement Learning for Large-Scale Distributed Multirobot Coordination. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 665–676. <https://doi.org/10.1109/TNNLS.2025.14358>.
36. Wang, Y. Facilitating Multi-Agent Coordination Relying on Graph Information Representation. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 17929–17940. <https://doi.org/10.1109/TNNLS.2025.3575196>.
37. Ding, S.; Du, W.; Ding, L.; Zhang, J.; Guo, L.; An, B. Multiagent reinforcement learning with heterogeneous graph attention network. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 6851–6860. <https://doi.org/10.1109/TNNLS.2023.3243557>.
38. Qi, S.; Huang, X.; Peng, P.; Huang, X.; Zhang, J.; Wang, X. Cascaded Attention: Adaptive and Gated Graph Attention Network for Multiagent Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 3769–3779. <https://doi.org/10.1109/TNNLS.2022.3197918>.
39. Yang, Q.; Lu, J.; Zhang, Y.; Shao, S. Multi-UAV Cooperative Path Planning via GNN and PPO. *IEEE Access* **2025**, *13*, 193579. <https://doi.org/10.1109/ACCESS.2025.11230560>.
40. Xiong, Z.; Xi, Y.; Cao, Y.; Li, C.; Li, R.; Li, J. Task Allocation with Communication Coordination in UAV Swarms via Asynchronous Multi-Objective Policy Optimization. *Aerosp. Sci. Technol.* **2026**, *174*, 111760. <https://doi.org/10.1016/j.ast.2026.111760>.
41. Sheikh, H.U.; Bölöni, L. Multi-Agent Reinforcement Learning for Problems with Combined Individual and Team Reward. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. <https://doi.org/10.1109/IJCNN48605.2020.9206879>.
42. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*; Vancouver, BC, Canada, 30 April–3 May 2018.
43. Huang, S.; Ontañón, S. A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. In *The International FLAIRS Conference Proceedings*; Florida Online Journals: Gainesville, FL, USA, 2022; Volume 35. <https://doi.org/10.32473/flairs.v35i.130584>.
44. Safaoui, S.; Vinod, A.P.; Chakrabarty, A.; Quirynen, R.; Yoshikawa, N.; Di Cairano, S. Safe multi-agent motion planning under uncertainty for drones using filtered reinforcement learning. *IEEE Trans. Robot.* **2024**, *40*, 2529–2542. <https://doi.org/10.1109/TRO.2024.3387010>.
45. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3803–3810. <https://doi.org/10.1109/ICRA.2018.8460528>.
46. Sadeghi, F.; Levine, S. CAD2RL: Real Single-Image Flight Without a Single Real Image. In *Proceedings of Robotics: Science and Systems XIII*; Cambridge, MA, USA, 12–16 July 2017; Robotics: Science and Systems Foundation: San Francisco, CA, USA, 2017. <https://doi.org/10.15607/RSS.2017.XIII.034>.
47. Dhuheir, M.; Erbad, A.; Hamdaoui, B.; Belhaouari, S.; Guizani, M.; Vu, T. Multi-Agent Meta Reinforcement Learning for Reliable and Low-Latency Distributed Inference in Resource-Constrained UAV Swarms. *IEEE Access* **2025**, *13*, 103045–103059. <https://doi.org/10.1109/ACCESS.2025.3572036>.
48. Guerra, A.; Guidi, F.; Dardari, D.; Djuric, P. Reinforcement Learning for Joint Detection and Mapping using Dynamic UAV Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *60*, 2586–2601. <https://doi.org/10.1109/TAES.2023.3300813>.
49. Sun, R.; Yu, G.; Chen, P.; Zhou, B. Cooperative Right-of-Way Assignment for Autonomous Mining Trucks at Multiple Open-Pit Mine Intersections. *Front. Ind. Eng.* **2025**, *3*, 1611512. <https://doi.org/10.3389/fieng.2025.1611512>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.