sensors

MDPI

Article

Adaptive Multi-UAV Formation Navigation in Narrow Passages via Reinforcement Learning with Task-Decomposed Hierarchical Training

Jian Li, Huiliao Yang *, Dawei Wu and Liangdong Wen

College of Artificial Intelligence and Automation, Hohai University, Nanjing 211106, China
* Correspondence: huiliao.yang@hhu.edu.cn

Abstract

Multi-UAV formation navigation in narrow passages requires coordinated collision avoidance and adaptive formation adjustment under geometric constraints. This paper proposes a leader–follower framework based on task-decomposed hierarchical Soft Actor–Critic (SAC) training. The leader is first trained to establish a stable navigation reference, after which its policy is fixed for follower training. A heterogeneous GRU–CNN (G-C) perception network processes proprioceptive and LiDAR observations, while distance-dependent formation-reward weighting enables progressive formation compression and recovery without predefined switching rules. Experiments on four narrow-passage topologies included in training show that the proposed method achieves 100% task success with mean formation-tracking errors of 0.2734–0.4263 m in static environments, and 90–100% task success with errors of 0.3838–0.5374 m in dynamic environments. Comparisons with hierarchical Proximal Policy Optimization (PPO) and curriculum-assisted multi-agent reinforcement learning baselines demonstrate a more favorable balance between task success and formation-tracking accuracy, particularly under moving-obstacle interference. These results demonstrate the effectiveness of the proposed framework within the evaluated scenarios.

Keywords: narrow-passage navigation; soft actor–critic (SAC); multi-UAV formation; adaptive reward mechanism

1. Introduction

With recent advances in intelligent control technologies, cooperative formation flight of multiple UAVs has become a prominent research topic in aerospace and robotics [1]. Compared with single-UAV systems, multi-UAV formations can alleviate individual limitations in payload capacity, endurance, and sensing range, thereby improving task efficiency and fault tolerance through spatial distribution and information sharing [2]. Consequently, multi-UAV formations have shown strong potential in disaster rescue, large-scale logistics, environmental monitoring, and military reconnaissance [3,4].

Regarding formation frameworks, mainstream approaches include the leader–follower framework [5], the virtual structure method [6], and the behavior-based distributed control method [7]. The leader–follower framework is widely adopted because of its structural simplicity and scalability. However, conventional leader–follower methods typically assume a rigid or semi-rigid formation geometry in which followers maintain fixed relative positions with respect to the leader. This assumption becomes restrictive in narrow-passage environments, where the available free space decreases substantially. Under such geometric

Check for updates

Academic Editor: Egidio D'Amato

Received: 29 September 2026
Revised: 26 October 2026
Accepted: 13 November 2026
Published: 18 November 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.

constraints, preserving a wide triangular formation may lead to collisions or deadlocks. Therefore, the formation must be capable of adaptively reconfiguring its geometry, such as compressing laterally in narrow spaces and recovering when sufficient space becomes available, without relying on manually designed switching rules.

Existing studies have investigated search-based planning, multi-agent reinforcement learning, heterogeneous perception, and hierarchical decision-making for multi-UAV navigation and formation control. A detailed review of these methods is provided in Section 2. Nevertheless, their effectiveness has mainly been demonstrated in open or moderately cluttered environments. Under severe geometric constraints, the inherent conflict between global path progression and local formation maintenance becomes more pronounced, making it difficult to learn a distributed policy that simultaneously coordinates navigation, collision avoidance, and formation adaptation.

In particular, formation navigation in environments containing narrow passages and dynamic obstacles presents three major challenges:

(1) *Partial observability and perception bottlenecks*: In the absence of global environmental priors, UAVs primarily rely on local LiDAR measurements and proprioceptive states. These observations contain spatial geometric information and temporally correlated motion information, respectively, and their effective extraction and fusion remain critical to efficient policy learning.
(2) *High dimensionality and task coupling*: A swarm navigating narrow passages must simultaneously perform macro-level global navigation and micro-level formation adaptation. Flat end-to-end reinforcement learning architectures require concurrent exploration of these coupled objectives in a high-dimensional state–action space, which may result in poor convergence and low learning efficiency.
(3) *Rigidity of fixed-weight rewards*: Conventional static reward functions cannot accommodate dynamically changing task priorities. In spatially constrained regions, excessive emphasis on formation maintenance may restrict the maneuverability required for safe collision avoidance, resulting in oscillatory behavior, collisions, or policy degradation.

In this study, map-free navigation refers to policy execution without a preloaded global obstacle map, using local LiDAR and proprioceptive observations together with the available target and formation-state information. This setting should be distinguished from generalization to map topologies not encountered during training.

These challenges are closely coupled throughout narrow-passage formation navigation. During leader training, proprioceptive states and local LiDAR observations are encoded by the G-C network to form the policy representation, while the sector-based LiDAR reward guides the leader toward a stable and centered navigation trajectory. After the leader policy converges, its real-time state determines the followers' ideal formation positions, and the resulting formation deviations become part of the followers' proprioceptive observations. During follower training, the G-C network jointly represents the temporal evolution of formation deviations and the spatial distribution of nearby obstacles, while the adaptive reward continuously modulates the strength of formation constraints according to local geometric conditions. The leader-generated reference trajectory therefore provides the basis for follower state construction and reward optimization, linking heterogeneous perception, staged policy learning, and adaptive formation regulation within a unified decision process.

To address the aforementioned issues, this paper proposes a multi-UAV adaptive formation navigation method based on the Soft Actor–Critic (SAC) algorithm under a leader–follower architecture for narrow-passage environments. The main contributions are summarized as follows:

1. A spatiotemporal heterogeneous perception network (G-C) is developed to extract temporal features from proprioceptive states using GRU and spatial features from LiDAR

observations using CNN. The fused representation integrates UAV motion states, formation deviations, and local environmental information to support formation tracking and collision avoidance.

2. A task-attribute-based two-stage hierarchical training strategy is proposed to reduce task coupling. The leader is first trained independently to establish a stable navigation reference, after which its policy is frozen to guide follower training, thereby reducing the coupling among global navigation, local collision avoidance, and formation maintenance during policy learning.
3. Role-specific adaptive reward functions are designed, incorporating sector-based LiDAR guidance for the leader and distance-dependent continuous formation-reward weighting for the followers. By adaptively balancing formation maintenance and collision avoidance, the proposed mechanism enables progressive formation compression and recovery without requiring predefined formation-switching rules.

The proposed components are evaluated through a common multi-seed protocol that combines qualitative trajectory analysis with four primary quantitative studies, complemented by a separate scalability evaluation. Specifically, the complete hierarchical SAC framework is compared with a hierarchical Proximal Policy Optimization (PPO) variant and with curriculum-assisted Multi-Agent Soft Actor–Critic (MASAC) and Multi-Agent Proximal Policy Optimization (MAPPO) baselines under centralized training and decentralized execution (CTDE); the G-C encoder is examined through modality-specific ablations; the continuous formation-weighting mechanism is compared with fixed and threshold-switch alternatives; and the obstacle-interaction threshold d_{th} is subjected to sensitivity analysis. These four studies are intended to separate the effects of the training hierarchy, base policy optimizer, heterogeneous perception structure, and adaptive reward mechanism. The scalability evaluation additionally examines the task performance of four-UAV and five-UAV formations in both static and dynamic narrow-passage environments.

2. Related Work

Traditional search-based and heuristic planning methods have been extensively applied to multi-UAV formation navigation. Improved variants of the RRT* algorithm have been used to enhance cooperative planning efficiency [8], modified artificial potential field methods have been developed to alleviate local-minimum problems [9], and hybrid frameworks combining A^* with the dynamic window approach have been proposed to improve real-time navigation performance [10]. These methods generally provide favorable interpretability and stability in known or weakly dynamic environments. However, their performance often depends on accurate environmental maps, carefully tuned parameters, and predefined coordination or formation-switching rules, which limits their adaptability in unknown environments with strong geometric constraints [11].

Reinforcement learning provides an alternative approach by enabling UAVs to learn cooperative navigation policies through interaction with the environment [12,13]. Early value-based methods, such as Deep Q-Network (DQN) and Dueling Double Deep Q-Network (D3QN), have been applied to leader–follower control and collision-free path planning [14,15]. However, their discrete action representations may limit control resolution in continuous UAV maneuvering tasks. Actor–critic methods have therefore received increasing attention in multi-UAV systems. Multi-agent deep reinforcement learning has been employed for collision-avoiding flocking in obstacle-cluttered environments and for joint target assignment and path planning [16,17]. Zhao et al. [18] proposed the Multiagent Joint Proximal Policy Optimization (MAJPPO) algorithm for cooperative fixed-wing UAV formation and obstacle avoidance. Fang et al. [19] employed multi-agent Soft Actor–Critic (MASAC) for collaborative path planning, jointly considering navigation,

collision avoidance, and formation-related objectives. MAPPO variants have also been investigated for multi-UAV path planning and relay coordination because of their stable on-policy updates [20]. Meanwhile, SAC-based planners have demonstrated effective continuous-control exploration in unknown and densely obstructed environments [21,22]. These studies demonstrate the feasibility of applying multi-agent policy optimization and maximum-entropy reinforcement learning to cooperative UAV control.

As navigation and formation objectives become increasingly coupled, several studies have introduced structural mechanisms to improve learning efficiency. Recent research has further investigated reinforcement-learning-based formation control with static and dynamic obstacle avoidance, as well as two-timescale trajectory planning for formation missions in unknown environments [23,24]. Zhou et al. [25] proposed TD-MATD3, in which the path-planning problem is decomposed into a navigation module and an obstacle-avoidance module. The corresponding actor–critic networks are constructed separately according to the two task objectives, and the navigation features are subsequently introduced into the obstacle-avoidance module. This task-oriented decomposition improves convergence compared with directly optimizing the complete path-planning objective. Nevertheless, the decomposition is performed within a flat multi-agent policy and does not distinguish the functional roles of different UAVs in the formation. By contrast, the framework considered in this study organizes the learning process according to the task attributes of the leader and followers, and the converged leader policy provides a stable navigation trajectory and geometric reference for subsequent follower training.

In addition to policy and task decomposition, effective perception under partial observability is essential for multi-UAV cooperation. Su et al. [26] developed a recurrent-MAPPO framework for cooperative multi-UAV searching and tracking. In their network, CNN layers first extract spatial features from map-based observations, after which a GRU captures information contained in historical observations. Liu et al. [27] combined deep reinforcement learning with affine transformations for flexible formation control in unknown obstacle environments. Their model–data hybrid framework employs an end-to-end policy to map onboard LiDAR observations to collision-avoidance actions, while a separate affine-transformation layer explicitly generates formation translation, rotation, and scaling. These studies demonstrate that CNNs, GRUs, LiDAR perception, and geometric formation transformations can effectively improve UAV cooperation. In these representative approaches, CNN and GRU modules are generally applied sequentially to a unified observation representation, or LiDAR-based policies are combined with explicitly prescribed formation-transformation models.

The G-C network proposed in this study adopts a modality-specific organization of the observation space. Rather than processing all observations through a common sequential encoder, it separates the state according to its physical characteristics. The GRU branch extracts temporal dependencies from proprioceptive variables, including motion and formation-error states, whereas the CNN branch extracts local spatial structures from consecutive LiDAR observations. The two modality-specific representations are subsequently fused for policy learning, enabling the policy to jointly characterize the temporal evolution of formation deviations and the spatial constraints imposed by the surrounding environment.

Hierarchical reinforcement learning has also been investigated to alleviate the high-dimensional decision-making burden in multi-UAV systems. Conventional HRL frameworks generally decompose a control process according to different command-abstraction levels. For example, Pang et al. [28] proposed a leader–follower multi-agent PPO framework for multi-UAV cooperative combat. Their hierarchy consists of a top level for situation assessment and policy selection, a middle level for generating desired maneuver angles,

and a bottom level for producing continuous control commands. The leader–follower relationship is incorporated into the top-level policy to improve coordination among agents.

The hierarchical structure considered in this study differs from the above high-level/low-level action hierarchy in both its decomposition criterion and training procedure. Specifically, the cooperative task is decomposed according to the task attributes of the leader and followers. The leader is first trained independently to establish a stable global navigation reference. After convergence, the leader policy is frozen and introduced into the followers' training environment as a geometric reference. The followers are subsequently trained to maintain formation and perform local collision avoidance while retaining target-directed progression through their navigation reward. The resulting hierarchy therefore constitutes a task-attribute-based sequential training strategy rather than a multilevel command- or action-abstraction hierarchy.

To further clarify these methodological differences, representative methods for multi-UAV formation and cooperative navigation are compared in Table 1. The comparison focuses on the underlying methodological framework, perception and temporal modeling, and the mechanisms used for task decomposition and formation adaptation.

Table 1. Comparison of representative methods for multi-UAV formation and cooperative navigation, with emphasis on formation adaptation.

Method	Methodological Framework	Perception and Temporal Modeling	Coordination and Formation Adaptation
Fang et al. [19]	MASAC	Local observations	Joint optimization of navigation, avoidance, and formation maintenance
Zhou et al. [25]	TD-MATD3	Task-related state features	Navigation and obstacle-avoidance task decomposition
Pang et al. [28]	LFMAPPO	Multi-level state representation	Three-level leader–follower hierarchy
Liu et al. [27]	DRL with affine transformation	End-to-end LiDAR perception	Explicit affine translation, rotation, and scaling of the formation
Xing et al. [29]	Improved MATD3 with hierarchical DRL	LSTM-enhanced environmental perception	Adaptive formation maintenance and transformation in narrow passages
Bui et al. [30]	Distributed behavior-based control	Local onboard sensing	Automatic V-shape contraction, expansion, and line reconfiguration
Proposed method	SAC-based hierarchical framework	GRU–CNN heterogeneous dual-stream encoding	Role-oriented two-stage training and continuous formation-reward weighting for compression and recovery without predefined switching rules

Existing studies have advanced multi-UAV cooperative navigation through multi-agent reinforcement learning, task decomposition, hierarchical decision-making, and adaptive formation control. Fang et al. [19] employed MASAC to jointly optimize navigation, collision avoidance, and formation maintenance, while Zhou et al. [25] and Pang et al. [28] adopted task decomposition and hierarchical decision-making, respectively, to reduce the

complexity of cooperative learning. Regarding formation adaptation, Liu et al. [27] combined deep reinforcement learning with affine transformations, Xing et al. [29] employed an improved MATD3 algorithm for adaptive formation transformation, and Bui et al. [30] developed distributed behavior-based control for automatic formation reconfiguration in narrow spaces. These studies provide important foundations for cooperative navigation and flexible formation control in complex environments.

Nevertheless, in narrow passages with dynamic obstacles, the coupling among navigation, collision avoidance, and formation maintenance may still increase the difficulty of policy learning. Existing task-decomposition and hierarchical approaches primarily focus on navigation subtasks or decision-making levels, while some formation-reconfiguration methods rely on explicit geometric transformation models or predefined behavior mechanisms. Effectively incorporating the task attributes of different UAVs and continuously coordinating formation maintenance and collision avoidance according to local environmental constraints therefore remain important research challenges.

To address these challenges, this study adopts a task-attribute-based two-stage hierarchical training strategy to reduce the coupling between navigation and formation control, together with a G-C heterogeneous perception network that integrates proprioceptive and LiDAR observations. Furthermore, distance-dependent continuous adaptive formation-reward weighting coordinates formation maintenance and collision avoidance, enabling the learned policy to achieve progressive formation compression and recovery without predefined formation-switching rules.

3. Problem Description and Modeling

3.1. UAV Kinematics Modeling

For multi-UAV cooperative navigation and formation keeping in narrow-passage environments, formations commonly maintain an approximately constant altitude during cruise to simplify coordination and avoid unnecessary vertical maneuvers [31]. Accordingly, this study represents the system's kinematic evolution and formation geometry in a two-dimensional horizontal plane, with particular emphasis on lateral obstacle avoidance and formation reconfiguration in narrow passages. The control process is described at the navigation level through linear acceleration and angular velocity, while attitude response and low-level flight-control dynamics are abstracted by the corresponding kinematic constraints. The resulting model therefore characterizes constant-altitude formation navigation and geometric adjustment in the horizontal plane. All members of the proposed multi-UAV system are homogeneous UAVs: the leader UAV (UAV^L) and the follower UAVs (UAV_i^F) differ only in task-level roles, with no hardware differences (i.e., all UAVs share an identical kinematic model). At discrete time step t, the state of each UAV is described by its absolute position (x, y) in the inertial coordinate system, its linear velocity v_t, and its heading angle ψ_t (Figure 1). The control inputs are linear acceleration $a_{c,t}$ and angular velocity ω_t. Based on Euler discretization, the evolution of the UAV's position, velocity, and heading can be described by the following discrete-time nonlinear kinematic equations:

$$\begin{cases} x_{t+1} = x_t + v_t \cos(\psi_t)\Delta t \\ y_{t+1} = y_t + v_t \sin(\psi_t)\Delta t \\ v_{t+1} = v_t + a_{c,t}\Delta t \\ \psi_{t+1} = \psi_t + \omega_t \Delta t \end{cases} \tag{1}$$

Here, $\Delta t > 0$ denotes the fixed sampling interval (time step) used in the discrete-time kinematic update.

Within the adopted kinematic description, the admissible control inputs and motion states are bounded by the operating region $\mathcal{F}$, which is defined by the following constraints:

1. Linear acceleration and angular velocity constraints: The linear acceleration and angular velocity are strictly limited within the feasible region, i.e., $a_{c,t} \in [a_{c,\min}, a_{c,\max}]$ and $\omega_t \in [\omega_{\min}, \omega_{\max}]$;
2. Flight speed and heading constraints: The flight speed v_t should satisfy $v_t \in [v_{\min}, v_{\max}]$, and the heading ψ_t should also satisfy $\psi_t \in [\psi_{\min}, \psi_{\max}]$.

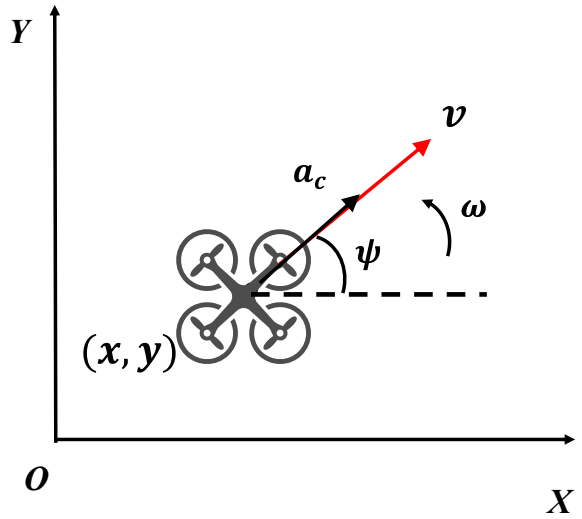


Figure 1. Kinematic model of the UAV.

3.2. Leader–Follower Task Modeling

From a macroscopic perspective, the execution of multi-UAV cooperative missions fundamentally relies on the following core objectives: first, ensuring collision-free flight for every UAV within the swarm. By modeling each UAV with a protective bounding circle of safe collision radius d_{safe}, this objective requires that the distance between any two UAVs i and j satisfies $d_{ij} \geq 2d_{\text{safe}}$, and the distance between any UAV and obstacles satisfies $d_{\text{obs}} \geq d_{\text{safe}}$; second, promoting target-directed navigation by minimizing the cumulative goal-distance cost J_D; and third, maintaining specific formation coordination requirements, such as a stable distributed communication network or spatial task distribution, by minimizing the formation distance errors J_E of all UAVs.

In the simplified two-dimensional simulation model, each UAV is represented by a circular body with an equivalent radius of $r_{\text{UAV}} = 0.5\,\text{m}$, which is equal to the safe collision radius d_{safe}. No additional safety margin is modeled separately.

If the multi-UAV system is treated as a monolithic whole to jointly solve for target progression, obstacle avoidance, and formation maintenance, the joint decision-making complexity increases substantially as these objectives become more strongly coupled. Therefore, to reduce the task difficulty, this paper adopts a leader–follower formation framework in which the leader is primarily responsible for minimizing the macro-navigation cost J_D and generating a stable navigation reference. The followers minimize the formation-maintenance cost J_E relative to this reference while retaining target-directed progression and local collision avoidance through their navigation reward. This task-attribute-based decomposition reduces the effective joint learning burden and provides a structured mechanism for coordinating global navigation and local formation regulation. It should be noted that the hierarchical decomposition itself does not constitute an unconditional guarantee of collision-free navigation. The formation-error boundedness, recovery, and associated

geometric safety implications are analyzed in Section 3.3. Furthermore, the following assumptions are made for the proposed multi-UAV system:

Assumption 1 (State and target availability)**.** *The target position is known in the global coordinate frame. At each control step, each UAV has access to its position, speed, and heading, from which the target-relative navigation variables are constructed.*

Assumption 2 (Sensing and information exchange)**.** *Each UAV is equipped with onboard LiDAR of identical measurement characteristics and receives local range observations. Formation members exchange the motion states and relative information required for cooperative decision-making at the common control rate.*

Based on the above assumptions, the navigation task of the leader UAV^L is defined as minimizing the cumulative goal-distance cost to the target point $\mathbf{p}^t(x^t, y^t)$ over the entire time horizon T, subject to the physical boundaries $\mathcal{F}$ defined in Section 3.1:

$$\min J_D = \sum_{t=0}^{T} \|\mathbf{p}^L(t) - \mathbf{p}^t\|_2 \tag{2}$$

where $\mathbf{p}^L(t)$ is the real-time position of the leader at time t.

Once the leader's navigation task is determined, the followers regulate their local formation relationship with respect to the leader while continuing target-directed progression. The desired relative geometry is defined as shown in Figure 2. Let the desired formation distance of any follower UAV_i^F be l_i, and the relative angle between the ideal position vector and the leader's heading be α_i. Based on the leader's real-time state (x_t^L, y_t^L, ψ_t^L), the ideal reference coordinates $\mathbf{p}_{i,t}^I(x_{i,t}^I, y_{i,t}^I)$ of the follower in the desired formation can be calculated through the following affine transformation:

$$\begin{cases} x_{i,t}^I = x_t^L - l_i \cos(\psi_t^L + \alpha_i) \\ y_{i,t}^I = y_t^L - l_i \sin(\psi_t^L + \alpha_i) \end{cases} \tag{3}$$

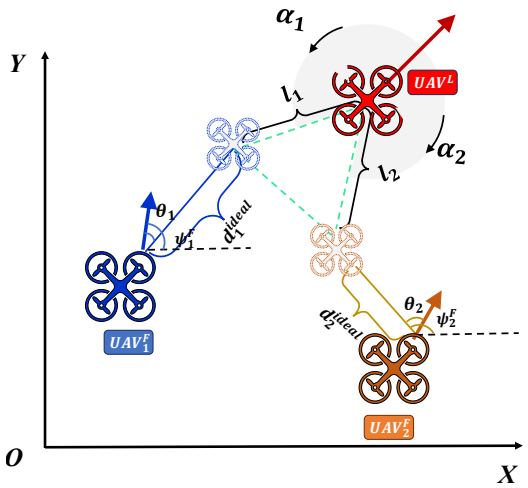


Figure 2. UAV formation geometry.

To quantitatively characterize the micro-coordination precision of the follower, the Euclidean spatial distance error $d_{i,t}^{\text{ideal}}$ of the i-th follower at time t relative to its ideal position is defined:

$$d_{i,t}^{\text{ideal}} = \sqrt{(x_{i,t}^F - x_{i,t}^I)^2 + (y_{i,t}^F - y_{i,t}^I)^2} \tag{4}$$

The heading-offset angle is defined as the angle between the follower's current heading vector and the vector pointing from its position to the ideal position, calculated as follows:

$$\theta_{i,t}^{\text{raw}} = \text{atan2}\Big(y_{i,t}^{I} - y_{i,t}^{F},\, x_{i,t}^{I} - x_{i,t}^{F}\Big) - \psi_{i,t}^{F}$$

$$\theta_{i,t} = \begin{cases} \theta_{i,t}^{\text{raw}} - 2\pi, & \text{if } \theta_{i,t}^{\text{raw}} > \pi \\ \theta_{i,t}^{\text{raw}} + 2\pi, & \text{if } \theta_{i,t}^{\text{raw}} \leq -\pi \\ \theta_{i,t}^{\text{raw}}, & \text{otherwise} \end{cases} \tag{5}$$

where $\psi_{i,t}^{F}$ denotes the follower heading measured from the positive x-axis.

Subject to the physical boundaries $\mathcal{F}$ defined in Section 3.1, J_E is defined as minimizing the sum of the errors $d_{i,t}^{\text{ideal}}$ between all followers UAV_i^F and their corresponding ideal positions over the entire time horizon T:

$$\min J_E = \sum_{i=1}^{n} \sum_{t=0}^{T} d_{i,t}^{\text{ideal}} \tag{6}$$

Here, n denotes the number of follower UAVs, and T is the final discrete time index of the navigation horizon.

3.3. Theoretical Analysis of Formation Boundedness and Recovery

Since the follower policies in this study are learned through reinforcement learning rather than derived from an explicit analytical feedback controller, global convergence of the SAC training process is not assumed. Instead, the following analysis characterizes the boundedness and recoverability of the formation error under the kinematic constraints defined in Section 3.1. The motion bounds provide a finite-horizon estimate of formation error, whereas finite-time recovery additionally requires a local closed-loop condition that SAC training does not inherently guarantee. For the analysis, Equation (1) is viewed through its underlying continuous-time kinematics, whose Euler discretization gives the model used in simulation:

$$\dot{\mathbf{p}} = v \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix}, \qquad \dot{\psi} = \omega. \tag{7}$$

To cover the generally asymmetric angular-velocity interval defined in Section 3.1, let $\bar{\omega} \triangleq \max\{|\omega_{\min}|, |\omega_{\max}|\}$ denote the maximum admissible angular-velocity magnitude. Thus, $|\omega(t)| \leq \bar{\omega}$.

For follower i, define the formation tracking error as

$$\mathbf{e}_i(t) = \mathbf{p}_i^F(t) - \mathbf{p}_i^I(t), \qquad d_i(t) = \|\mathbf{e}_i(t)\|_2, \tag{8}$$

where $\mathbf{p}_i^I(t)$ is the ideal formation position defined in Equation (3), and $d_i(t)$ corresponds to the formation-distance error $d_{i,t}^{\text{ideal}}$ defined in Equation (4).

Lemma 1 (Bounded motion of the ideal formation reference)**.** *Under the kinematic constraints $v^L \leq v_{\max}$ and $|\omega^L| \leq \bar{\omega}$, the motion of the ideal formation position satisfies*

$$\|\dot{\mathbf{p}}_i^I(t)\|_2 \leq v_{\max} + l_i\bar{\omega}. \tag{9}$$

Consequently, since $\|\dot{\mathbf{p}}_i^F(t)\|_2 \leq v_{\max}$, $d_i(t)$ is absolutely continuous and its rate of variation satisfies, for almost every t,

$$|\dot{d}_i(t)| \leq 2v_{\max} + l_i\bar{\omega} \triangleq B_i. \tag{10}$$

Proof of Lemma 1. Differentiating the ideal-position relation in Equation (3) gives

$$\dot{\mathbf{p}}_i^I = \dot{\mathbf{p}}^L - l_i \omega^L \begin{bmatrix} -\sin(\psi^L + \alpha_i) \\ \cos(\psi^L + \alpha_i) \end{bmatrix}. \tag{11}$$

Applying the triangle inequality and the speed and angular-velocity bounds yields Equation (9). Since the Euclidean norm is Lipschitz continuous, $d_i(t)$ is absolutely continuous and, for almost every t,

$$|\dot{d}_i(t)| \le \|\dot{\mathbf{p}}_i^F(t) - \dot{\mathbf{p}}_i^I(t)\|_2 \le 2v_{\max} + l_i\bar{\omega}, \tag{12}$$

which proves Equation (10). □

Assumption 3 (Local formation-recovery condition)**.** *After an obstacle-induced formation reconfiguration, suppose that the follower remains in a region where* $d_{\min}^{\text{obs}}(t) \ge d_{\text{th}}$ *and hence* $W_{\text{formation}} = 1$ *according to Equation (36). In this region, whenever* $d_i(t) > d_i^r$*, the learned closed-loop follower policy is assumed to provide a positive relative closing speed toward the ideal formation position satisfying*

$$v_i^F \cos\theta_i - \mathbf{q}_i^T \dot{\mathbf{p}}_i^I \ge \nu_i > 0, \qquad \mathbf{q}_i = \frac{\mathbf{p}_i^I - \mathbf{p}_i^F}{\|\mathbf{p}_i^I - \mathbf{p}_i^F\|_2}, \tag{13}$$

where $\mathbf{q}_i$ *is the unit vector pointing from the follower toward its ideal formation position,* $d_i^r > 0$ *is the prescribed tracking tolerance, and* ν_i *denotes a positive relative closing-speed margin. Assumption 3 is an additional sufficient condition for local closed-loop recovery, not a property guaranteed by SAC training or the adaptive reward mechanism. In particular,* $W_{\text{formation}} = 1$ *does not by itself imply that the learned policy satisfies Equation (13).*

Theorem 1 (Finite-Horizon Formation-Deviation Bound and Conditional Recovery)**.** *Let* $[t_a, t_b]$ *denote the interval during which follower i undergoes obstacle-induced formation relaxation or reconfiguration, with* $t_b - t_a \le \tau_i$*. If the formation error satisfies* $d_i(t_a) \le d_i^r$ *before reconfiguration, then throughout this interval*

$$d_i(t) \le d_i^r + B_i\tau_i \triangleq \bar{d}_i, \qquad t \in [t_a, t_b]. \tag{14}$$

Furthermore, if Assumption 3 holds after t_b *until the error returns to* d_i^r*, then the formation error recovers to this tolerance in finite time.*

Proof of Theorem 1. From Lemma 1, $|\dot{d}_i(t)| \le B_i$ for almost every t. Integrating over $[t_a, t]$ gives

$$d_i(t) - d_i(t_a) \le B_i(t - t_a) \le B_i\tau_i, \tag{15}$$

and Equation (14) follows from $d_i(t_a) \le d_i^r$. If $d_i(t_b) \le d_i^r$, the recovery time is zero. Otherwise, while $d_i(t) > d_i^r$, the error dynamics satisfy

$$\dot{d}_i(t) = \mathbf{q}_i^T \dot{\mathbf{p}}_i^I - v_i^F \cos\theta_i. \tag{16}$$

Using Assumption 3 yields

$$\dot{d}_i(t) \le -\nu_i < 0. \tag{17}$$

Therefore, the error decreases monotonically after the reconfiguration and returns to d_i^r within a finite recovery time satisfying

$$T_i^{\mathrm{rec}} \le \frac{[d_i(t_b) - d_i^r]_+}{\nu_i} \le \frac{B_i \tau_i}{\nu_i}. \tag{18}$$

Here, $[x]_+ = \max\{x, 0\}$. Thus, the formation deviation remains bounded during the finite obstacle-induced reconfiguration interval under the stated motion and initial-error constraints. Subsequent finite-time recovery is guaranteed only if the local closed-loop condition in Assumption 3 continues to hold until the tracking tolerance is reached. □

For the geometric safety analysis, let $\mathcal{O}_t$ denote the obstacle-occupied region at time t, and let $d_{\mathcal{O}_t}(\mathbf{p}) = \inf_{\mathbf{o} \in \mathcal{O}_t} \|\mathbf{p} - \mathbf{o}\|_2$ denote the minimum Euclidean distance from position $\mathbf{p}$ to this region.

Corollary 1 (Geometric safety implication under bounded formation error)**.** *If the formation errors satisfy $d_k(t) \le \bar{d}_k$ for every follower k and, in a region where the nominal formation geometry is feasible, the following geometric margins hold for each follower i and every distinct pair $i \ne j$,*

$$\begin{aligned} d_{\mathcal{O}_t}(\mathbf{p}_i^I) &\ge d_{\mathrm{safe}} + \bar{d}_i, \\ l_i &\ge 2d_{\mathrm{safe}} + \bar{d}_i, \\ \|\mathbf{p}_i^I - \mathbf{p}_j^I\|_2 &\ge 2d_{\mathrm{safe}} + \bar{d}_i + \bar{d}_j, \qquad i \ne j, \end{aligned} \tag{19}$$

then the actual follower positions satisfy

$$\begin{aligned} d_{\mathcal{O}_t}(\mathbf{p}_i^F) &\ge d_{\mathrm{safe}}, \\ \|\mathbf{p}_i^F - \mathbf{p}^L\|_2 &\ge 2d_{\mathrm{safe}}, \\ \|\mathbf{p}_i^F - \mathbf{p}_j^F\|_2 &\ge 2d_{\mathrm{safe}}. \end{aligned} \tag{20}$$

Proof of Corollary 1. At each time t, the distance-to-obstacle-set function is 1-Lipschitz. Combining this property with the reverse triangle inequality and the ideal formation geometry in Equation (3) yields the following bounds:

$$\begin{aligned} d_{\mathcal{O}_t}(\mathbf{p}_i^F) &\ge d_{\mathcal{O}_t}(\mathbf{p}_i^I) - d_i, \\ \|\mathbf{p}_i^F - \mathbf{p}^L\|_2 &\ge \|\mathbf{p}_i^I - \mathbf{p}^L\|_2 - d_i = l_i - d_i, \\ \|\mathbf{p}_i^F - \mathbf{p}_j^F\|_2 &\ge \|\mathbf{p}_i^I - \mathbf{p}_j^I\|_2 - d_i - d_j. \end{aligned} \tag{21}$$

Substituting $d_i \le \bar{d}_i$, $d_j \le \bar{d}_j$, and the margins in Equation (19) proves Equation (20). □

The geometric safety implication applies only where the tracking-error bounds and nominal clearance margins are satisfied; it does not establish collision avoidance where the nominal formation geometry is infeasible.

Remark 1. *The finite-horizon formation-error bound in Theorem 1 follows from the kinematic constraints, the initial-error condition, and the bounded reconfiguration duration. In contrast, its finite-time recovery conclusion additionally requires Assumption 3 to hold throughout the recovery interval. The geometric safety implication in Corollary 1 also depends on the stated error bounds and nominal clearance margins.*

Because the follower policies are learned using SAC rather than analytically constructed to enforce the recovery inequality, the training process does not establish that the learned policies satisfy Assumption 3. The simulations in Section 5.3 illustrate formation compression and recovery in the evaluated scenarios, providing empirical evidence of recovery behavior rather than a mathematical verification of Assumption 3 in all states. Accordingly, the results are conditional

analytical statements, not unconditional guarantees of SAC convergence, formation recovery, or collision avoidance.

4. Hierarchical Formation Training Framework for Narrow Passages

The coupled objectives established in Section 3, namely the leader's macro-navigation cost J_D defined by Equation (2) and the followers' formation-maintenance cost J_E defined by Equation (6), constitute a challenging multi-objective decision-making problem. In narrow passages, partial observability and strong spatial coupling complicate real-time navigation and formation adjustment when obstacle geometry is perceived through local sensing without a preloaded global obstacle map. Therefore, this paper maps the multi-UAV cooperative decision-making process into a Reinforcement Learning (RL) paradigm.

To systematically address the three challenges outlined in Section 1 (perception bottlenecks, coupled decision-making, and rigid reward design), we propose a hierarchical training framework illustrated in Figure 3. The framework consists of two sequential stages:

- **Stage 1 (Leader Pre-training)**: Only the leader UAV is trained. Its state s_L, defined in Section 4.1, is encoded by the G-C network described in Section 4.3 to obtain the feature representation o_L. The leader policy network then takes o_L as input and outputs the continuous action a_L defined in Section 4.2. The environment returns the leader reward R_L defined in Section 4.4, which consists of the navigation reward with sector-based LiDAR avoidance and goal-oriented progression. The policy network is trained using the SAC procedure described in Section 4.5. After convergence, the trained leader policy π_L is obtained.
- **Stage 2 (Followers Team Training)**: Unlike Stage 1, the pre-trained leader policy π_L is frozen and no longer updated. Instead, it is imported into the training environment to generate a stable geometric reference trajectory for the followers. Each follower, with its own state s_F^i (which includes additional formation-related terms; see Section 4.1), is processed through the same G-C encoder to yield the feature representation o_F^i. The resulting feature representation is fed into the follower policy network, which is trained to learn formation maintenance and local collision avoidance. The followers are also trained using the same SAC algorithm, but receive the follower-specific reward R_F^i defined in Section 4.4, which combines the navigation and adaptive formation-reward terms to balance formation tracking and obstacle avoidance.

This two-stage hierarchical design first stabilizes the global navigation reference and then optimizes follower formation regulation and local collision avoidance, thereby reducing task coupling during policy learning. The detailed components of the proposed framework will be introduced in the remainder of this section.

4.1. State Space Design

When dynamic obstacles are perceived through local observations, partial observability makes it difficult for UAVs to construct a sufficiently informative state representation from a single sensing modality. To address this issue, this paper explicitly constructs a dual-stream state space composed of proprioceptive states and exteroceptive states within the reinforcement learning framework. This design simultaneously characterizes the UAV's own motion state, formation geometric relationships, and the spatial topological structure of the local environment. The state representation not only serves the decision-making requirements of different roles under the hierarchical control framework, but also provides a clear semantic boundary for the structural division of the subsequent heterogeneous perception network.

The proprioceptive state branch (Proprioceptive Stream), shown in Figure 4a, is mainly used to describe the UAV's kinematic state and its relative navigation relationship.

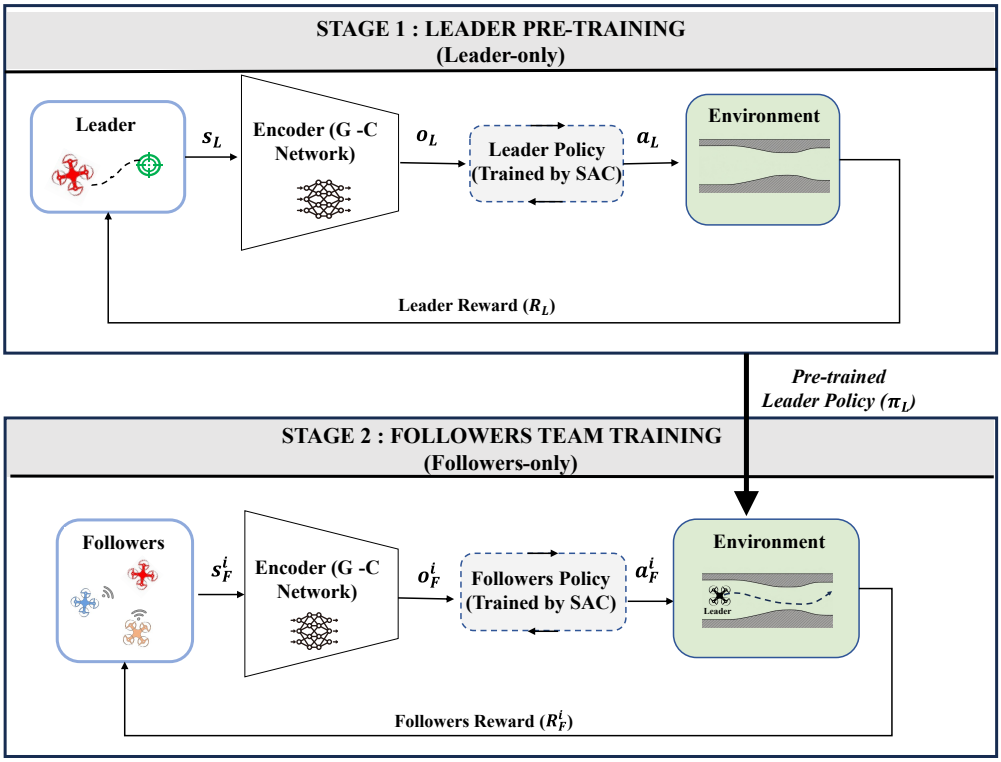
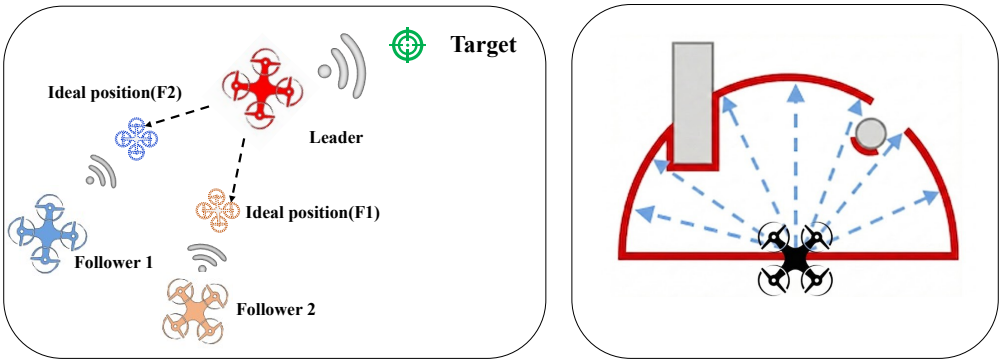


Figure 3. Hierarchical training framework.

For the leader UAV, whose decision objective focuses on global path progression and obstacle guidance, the proprioceptive state vector is defined as

$$s_P^L = [d_{\text{target}},\ \phi_t,\ v_t]^T \tag{22}$$

where d_{target} represents the relative distance between the leader's current position and the target point, ϕ_t denotes the target bearing angle, and v_t is the current flight speed. These state variables collectively characterize the instantaneous navigation status of the leader in the target coordinate frame, providing necessary information for generating a continuous and smooth reference trajectory.



(a) Proprioceptive stream **(b) Exteroceptive stream**

Figure 4. State space branches.

Compared with the leader, each follower UAV must not only accomplish its own navigation task but also maintain the desired geometric relationship with the leader. Therefore, its proprioceptive state must further reflect formation deviation and local conflict information. Based on the formation geometric model established in Section 3.2, the proprioceptive state of follower i is defined as

$$s_P^F = [d_{\text{target}},\ \phi_t,\ v_t,\ d_i^{\text{ideal}},\ \theta_i,\ d_{\text{others}}]^T \tag{23}$$

where d_i^{ideal} represents the Euclidean distance error between follower i and its ideal formation position (see Equation (4)), and θ_i denotes the heading deviation angle (see Equation (5)), used to measure the angular difference between its current heading and the direction toward the ideal position. The scalar feature d_{others} is taken as the distance from follower i to its nearest other formation member. Specifically, for a formation member i, define $d_i^{\text{other}} = \min_{j \neq i} \|\mathbf{p}_i - \mathbf{p}_j\|_2$, where $\mathbf{p}_i$ and $\mathbf{p}_j$ are current UAV positions; thus, for follower i, $d_{\text{others}} = d_i^{\text{other}}$. This is also the inter-UAV distance used in the collision condition of Section 4.4.1. These geometric quantities provide the follower policy with information about formation deviation and local inter-UAV relationships, thereby supporting formation regulation and collision avoidance.

In addition to the proprioceptive state, this paper further introduces an exteroceptive state branch (Exteroceptive Stream), shown in Figure 4b, to characterize the UAV's perception of the local environmental geometric structure. The exteroceptive state is obtained from onboard LiDAR measurements. Taking the UAV's current heading as the 0° reference direction, uniform angular sampling is performed within a $\pm 90^\circ$ field of view. Let the sampling resolution be 3°, resulting in a total of $N = 60$ distance measurement rays per frame. For the k-th ray, if an obstacle is detected within the maximum sensing radius $R_{\max}$, the measured distance d_k^{obs} is returned; otherwise, $R_{\max}$ is returned. Accordingly, the exteroceptive state vector is defined as

$$s_E = [d_1^{\text{obs}},\ d_2^{\text{obs}},\ \ldots,\ d_N^{\text{obs}}]^T \tag{24}$$

By integrating the above two sources of information, the complete state representation of each UAV is formed by concatenating the proprioceptive state and the exteroceptive state along the feature dimension, yielding the final state vectors $s_L = [(s_P^L)^T,\ (s_E^L)^T]^T$ and $s_F = [(s_P^F)^T,\ (s_E^F)^T]^T$. In this representation, position, speed, and heading are supplied by the onboard state-estimation process, whereas LiDAR ranges provide local exteroceptive observations. For each follower, the ideal position is computed from the leader state and the prescribed formation geometry, while the formation error and inter-UAV distances are constructed from the corresponding relative positions. The policy state therefore combines self-motion, local environmental geometry, and formation relationships within a unified representation.

With the short-sequence window $T_{\text{seq}} = 3$ (distinguished from the navigation horizon T), the leader has a three-dimensional proprioceptive state and a 60-dimensional LiDAR observation per time step, resulting in sequence inputs of 3×3 and 3×60 (189 input scalars in total). For each follower, d_{others} is a scalar feature, giving a six-dimensional proprioceptive state; the corresponding sequence inputs are 3×6 and 3×60 (198 input scalars in total). The two input sequences are processed separately by the GRU and CNN branches rather than concatenated before encoding.

4.2. Action Space Design

Considering that the control inputs in the UAV kinematic model (see Section 3.1) vary continuously in both time and magnitude, modeling the action space in continuous form can avoid the loss of control resolution and unnecessary approximation errors caused by discretization. Accordingly, the policy network parameterizes a two-dimensional Gaussian distribution for the unconstrained action $\tilde{a}$, with components corresponding to linear acceleration and angular velocity before action squashing:

$$\pi_{\text{pre}}(\tilde{a} \mid s) = \mathcal{N}\Big(\mu(s), \text{diag}(\sigma(s)^2)\Big) \tag{25}$$

Here, $\mu(s) \in \mathbb{R}^2$ and $\sigma(s) \in \mathbb{R}^2_{>0}$ are the state-dependent mean and component-wise standard-deviation vectors, respectively; $\text{diag}(\sigma(s)^2)$ denotes the diagonal covariance matrix.

The unconstrained action is then sampled using the reparameterization trick [32]:

$$\tilde{a} = \mu(s) + \sigma(s) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, I_2) \tag{26}$$

The symbol $\odot$ denotes element-wise multiplication, I_2 is the 2×2 identity matrix, and ε is a two-dimensional standard Gaussian noise vector.

Subsequently, a hyperbolic tangent function is applied to the unconstrained action and the normalized output is linearly mapped to the feasible physical control range of the UAV [33]. Let $a_{\min}$ and $a_{\max}$ denote the lower and upper action bounds applied component-wise to $[a_c, \omega]^T$. The physical action is obtained as

$$a = a_{\min} + \frac{\tanh(\tilde{a}) + \mathbf{1}_2}{2} \odot (a_{\max} - a_{\min}). \tag{27}$$

Here, $\mathbf{1}_2 = [1, 1]^T$ denotes the two-dimensional all-ones vector. Thus, $\pi_{\text{pre}}(\tilde{a} \mid s)$ describes the Gaussian distribution before squashing, whereas $\pi(a \mid s)$ denotes the resulting policy over bounded physical actions as used in the SAC objective.

4.3. Network Architecture Design

Based on the heterogeneous proprioceptive–exteroceptive state space constructed in Section 4.1, this paper designs a GRU-CNN spatiotemporal heterogeneous perception network, shown in Figure 5.

The overall network consists of three stages: heterogeneous feature encoding, feature fusion, and decision output. Its core idea lies in adopting dedicated feature extraction mechanisms according to the temporal characteristics and spatial structural properties of different state information, rather than applying a unified mapping to multimodal features.

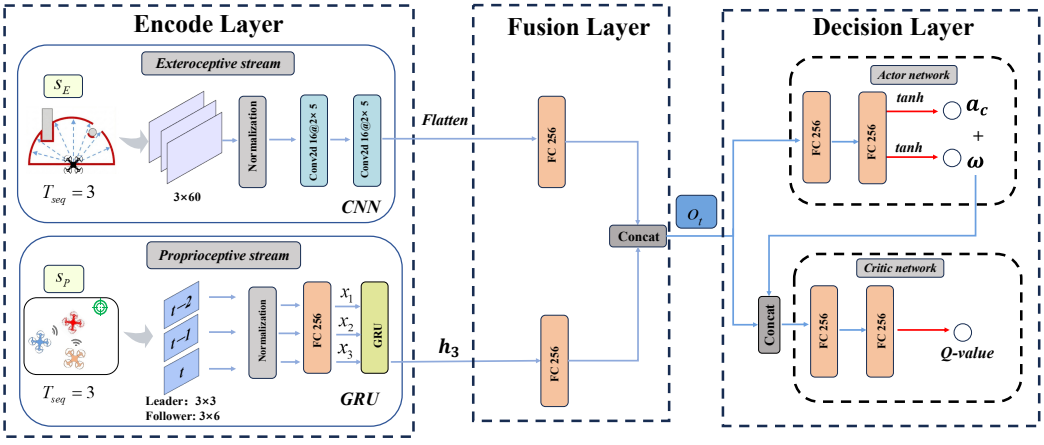


Figure 5. Encoder architecture.

The proprioceptive state branch mainly includes the UAV's kinematic state, relative navigation information, and formation geometric errors, which evolve over time and exhibit significant temporal correlation. To capture such dynamic characteristics, this paper adopts a Gated Recurrent Unit (GRU) as the proprioceptive feature encoder. Compared to memoryless networks like Multi-Layer Perceptron (MLP), the GRU leverages its internal gating mechanisms to selectively retain crucial historical kinematic trends and effectively

mitigate the vanishing gradient problem. This inherent advantage enables the network to form a more stable and continuous temporal representation of the formation's dynamic variations, which is essential for smooth trajectory tracking. Specifically, let the short temporal sequence input of proprioceptive states be $\{x_1, x_2, x_3\}$. The GRU updates the hidden state recursively according to $h_t = GRU(x_t, h_{t-1})$, and the hidden state at the last time step h_3 is taken as the proprioceptive feature representation.

The exteroceptive state branch is composed of LiDAR distance measurements, which are used to describe the spatial distribution characteristics of the environment surrounding the UAV. Considering that such information exhibits a high-dimensional spatial structure within a single frame, this paper employs a Convolutional Neural Network (CNN) to encode consecutive LiDAR observations. Through its local receptive fields and weight-sharing mechanisms, the CNN inherently excels at extracting local geometric topologies (such as obstacle boundaries and narrow passage contours) directly from raw, high-dimensional distance arrays. This structural advantage significantly reduces parameter redundancy while providing an effective spatial feature representation for the UAVs.

Considering the moderate dimensionality of the proprioceptive states and the 60-ray LiDAR observations, a relatively compact network configuration is adopted to avoid unnecessary model complexity. As illustrated in Figure 5, the CNN branch contains two convolutional layers with 16 kernels of size 2×5, while the encoded features are mapped through 256-dimensional fully connected layers before actor–critic decision making. A short sequence length of $T_{\text{seq}} = 3$ is used to introduce recent temporal information into both state streams without substantially increasing the input and computational burden. These architectural settings are kept fixed throughout all experiments to maintain a consistent network capacity across the comparative evaluations.

After completing the independent encoding of proprioceptive and exteroceptive features, the two types of features are first mapped to a unified dimension and then concatenated in the feature space. Subsequently, a multi-layer perceptron is applied to form a latent state representation for decision-making. This latent representation is used as a shared input and is fed into both the Actor network and the Critic network for continuous action generation and state–action value evaluation, respectively, thereby forming a complete actor–critic decision framework. The proprioceptive features directly serve the computation of formation-related reward terms, while the exteroceptive features provide perceptual support for environment-constrained reward modulation. This consistency in design ensures coordinated interaction among state modeling, network architecture, and policy optimization objectives. To verify the substantive modeling role of the above network structure, Section 5.4.1 presents systematic ablation experiments evaluating the performance impact of different encoder structures in the proprioceptive and exteroceptive branches.

4.4. Reward Function Design

In multi-UAV cooperative formation tasks, the reward function is not only used to characterize the task objectives but also directly influences the behavior pattern of the formation under complex geometrically constrained environments. According to the different task divisions between the leader and the followers in the formation, the structures of their reward functions also differ: the leader's reward function consists of the navigation reward, i.e., $R_L = R_{\text{navigation}}$, while the follower's reward function consists of both the navigation reward and the formation reward, i.e., $R_F = R_{\text{navigation}} + R_{\text{formation}}$. The specific forms of the navigation reward and the formation reward will be given below.

4.4.1. Navigation Reward

The navigation reward applies to both the leader and the followers, ensuring that the system progresses toward the target while avoiding collisions. Its overall form is defined as

$$R_{\text{navigation}} = r_{\text{step}} + r_{\text{avoid}} + r_{\text{goal}}. \tag{28}$$

In Equation (28), r_{avoid} denotes the role-specific obstacle-avoidance term: r^{L}_{avoid} for the leader and r^{F}_{avoid} for the followers, as defined below.

The term r_{step} describes the termination condition when the UAV reaches the target (i.e., $d_{\text{target}} \leq 1.0\,\text{m}$), or when a collision occurs (i.e., the minimum obstacle distance $d^{\text{obs}}_{\text{min}}$ detected by LiDAR is less than or equal to d_{safe}, or the inter-UAV distance d^{other}_{i} is less than or equal to $2d_{\text{safe}}$), as well as the normal step progression. It is defined as

$$r_{\text{step}} = \begin{cases} 100, & \text{if reach and no collision,} \\ -30, & \text{if collision,} \\ -0.2, & \text{otherwise.} \end{cases} \tag{29}$$

To encourage continuous progression toward the target, a goal-oriented reward based on distance variation is introduced:

$$r_{\text{goal}} = \frac{d^{t-1}_{\text{target}} - d^{t}_{\text{target}}}{v_{\max}\Delta t}, \tag{30}$$

where d^{t}_{target} denotes the distance between the UAV and the target at time step t, $v_{\max}$ is the specified maximum flight speed, and Δt is the decision time step.

Inspired by the LiDAR partition strategy in mobile robot navigation [33], the $\pm 90^{\circ}$ sensing range is divided into three sectors: left $(-90^{\circ}, -30^{\circ})$, center $(-30^{\circ}, 30^{\circ})$, and right $(30^{\circ}, 90^{\circ})$. The minimum obstacle distance in each sector is extracted separately.

For each LiDAR sector $k \in \{\text{left}, \text{center}, \text{right}\}$, let $d^{\text{obs}}_{\text{min},k}$ denote the minimum measured obstacle distance within that sector. The corresponding sector-specific avoidance penalty is defined as

$$r^{k}_{\text{avoid}} = \begin{cases} 2.0\left(e^{\left(\frac{1}{d_{\text{th}}} - \frac{1}{d^{\text{obs}}_{\text{min},k}}\right)} - 1.0\right), & d^{\text{obs}}_{\text{min},k} \leq d_{\text{th}}, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

Here, d_{th} is the obstacle-interaction threshold. Each sector penalty is zero at $d^{\text{obs}}_{\text{min},k} = d_{\text{th}}$ and decreases as an obstacle becomes closer within that sector.

The leader's overall avoidance term combines the three sector penalties $r^{\text{left}}_{\text{avoid}}$, $r^{\text{center}}_{\text{avoid}}$, and $r^{\text{right}}_{\text{avoid}}$ from Equation (31) as follows:

$$r^{L}_{\text{avoid}} = -\left| r^{\text{left}}_{\text{avoid}} - r^{\text{right}}_{\text{avoid}} \right| + r^{\text{center}}_{\text{avoid}}. \tag{32}$$

For the followers, we do not partition the LiDAR field of view into sectors. This is because, by referencing the leader's trajectory, the followers can more easily learn a center-aligned passage policy. Therefore, for the followers, $d^{\text{obs}}_{\text{min}}$ represents the distance to the nearest obstacle over the entire LiDAR field of view, and the followers' collision-avoidance penalty is defined as:

$$r_{\text{avoid}}^{F} = \begin{cases} 2.0\left(e^{\left(\frac{1}{d_{\text{th}}}-\frac{1}{d_{\min}^{\text{obs}}}\right)} - 1.0\right), & d_{\min}^{\text{obs}} \le d_{\text{th}}, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

The same d_{th} is used for both the leader and the followers so that obstacle interaction is defined on a common spatial scale throughout the hierarchical framework.

4.4.2. Formation Reward

Based on the aforementioned LiDAR sector-based obstacle-avoidance reward, this paper further considers the conflict between the formation-maintenance objective and safe obstacle avoidance in space-constrained environments. To prevent the overemphasis on formation constraints from restricting individual maneuverability in narrow passages, this paper designs the formation reward using the formation-distance error together with an adaptive weight adjustment:

$$R_{\text{formation}} = W_{\text{formation}} r_i^{\text{ideal}} \tag{34}$$

$$r_i^{\text{ideal}} = \begin{cases} (e^{(0.5-d_i^{\text{ideal}})} - 1.0), & \text{if} \quad d_i^{\text{ideal}} \le 5.0\,\text{m}, \\ -10, & \text{otherwise} \end{cases} \tag{35}$$

where r_i^{ideal} denotes the distance-error reward associated with the follower's ideal formation position, d_i^{ideal} is the formation-distance error defined in Section 3.2, and $W_{\text{formation}}$ is the adaptive weight of the formation constraint.

The 5.0 m cutoff in Equation (35) is used as a large-error boundary to distinguish ordinary formation-tracking deviations from severe formation loss; it is not used as a geometric switching threshold for obstacle avoidance or formation compression. The numerical offsets and scale factors in Equations (31), (33), and (35) are empirical reward-shaping constants selected to keep the magnitudes of different reward components within comparable ranges during training. These constants are kept fixed in all comparative and sensitivity experiments so that the influence of the obstacle-interaction threshold d_{th} can be isolated.

To enable the strength of the formation constraint to vary with environmental geometric conditions, this paper establishes a continuous mapping relationship between the formation weight $W_{\text{formation}}$ and the follower's minimum distance to surrounding obstacles $d_{\min}^{\text{obs}}$:

$$W_{\text{formation}} = \text{clip}\left(\frac{d_{\min}^{\text{obs}} - d_{\text{safe}}}{d_{\text{th}} - d_{\text{safe}}}, 0, 1\right), \qquad d_{\text{th}} > d_{\text{safe}}, \tag{36}$$

where $\text{clip}(\cdot)$ restricts the weight to the interval $[0, 1]$, d_{safe} is the prescribed safety distance, and d_{th} is the obstacle-interaction threshold introduced in Section 4.4.1. The same threshold is therefore shared by the obstacle-avoidance penalty and the adaptive formation-weight mechanism.

According to Equation (36), when $d_{\min}^{\text{obs}} \ge d_{\text{th}}$, $W_{\text{formation}} = 1$ and the formation constraint is fully retained. As the obstacle distance decreases from d_{th} toward d_{safe}, the formation weight decreases linearly from 1 toward 0, progressively reducing the influence of formation maintenance before the collision boundary is reached. At $d_{\min}^{\text{obs}} \le d_{\text{safe}}$, the collision termination condition defined in Equation (29) applies. Thus, entering the obstacle-interaction region simultaneously activates the avoidance penalty and progressively relaxes the formation constraint, providing a common spatial scale for coordinating the two objectives before collision termination.

Because the formation weight varies continuously with obstacle distance, this design avoids the abrupt 0–1 transition introduced by threshold-based hard switching and provides a smooth modulation of the relative priority between formation maintenance and collision avoidance. This continuous modulation supports progressive formation compression, recovery, and reconfiguration in geometrically constrained environments.

4.5. Hierarchical Training Strategy Based on SAC

Considering that the linear acceleration a_c and angular velocity ω vary continuously in both time and magnitude, reinforcement learning algorithms utilizing discretized action spaces (such as DQN and its variants) may induce control chattering and resolution loss. Continuous-control actor–critic algorithms are therefore more suitable for the present navigation problem. SAC [34] is adopted as the primary policy optimizer because its off-policy replay mechanism improves sample reuse, while entropy regularization encourages continued exploration in geometrically constrained environments. To separate the contribution of the task-attribute-based hierarchy from the choice of policy optimizer, Section 5.4.2 additionally evaluates the same hierarchical training structure with PPO; the proposed hierarchical SAC implementation is denoted HSAC in the comparative experiments.

Under the SAC framework, the optimization objective of the parameterized policy $\pi(a|s)$ is defined as finding the optimal policy π^* that maximizes the expected cumulative discounted return augmented with policy entropy:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t (r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)))\right] \tag{37}$$

where $r(s_t, a_t)$ is the immediate reward feedback from the environment (composed of macroscopic and microscopic objectives), and $\gamma \in (0, 1)$ is the discount factor. $\mathcal{H}(\pi(\cdot|s_t)) = -\mathbb{E}_{a_t \sim \pi(\cdot|s_t)}[\log \pi(a_t|s_t)]$ denotes the entropy of the policy given state s_t, reflecting the randomness in action selection. The temperature parameter α is utilized to dynamically balance the trade-off between reward maximization (exploitation) and entropy maximization (exploration).

Building upon the SAC algorithm, we design a task-attribute-based hierarchical training framework for the multi-UAV system (shown in Figure 6). Specifically, the leader is first trained independently to generate a stable navigation policy, and then the learned leader policy is integrated into the training process of the followers.

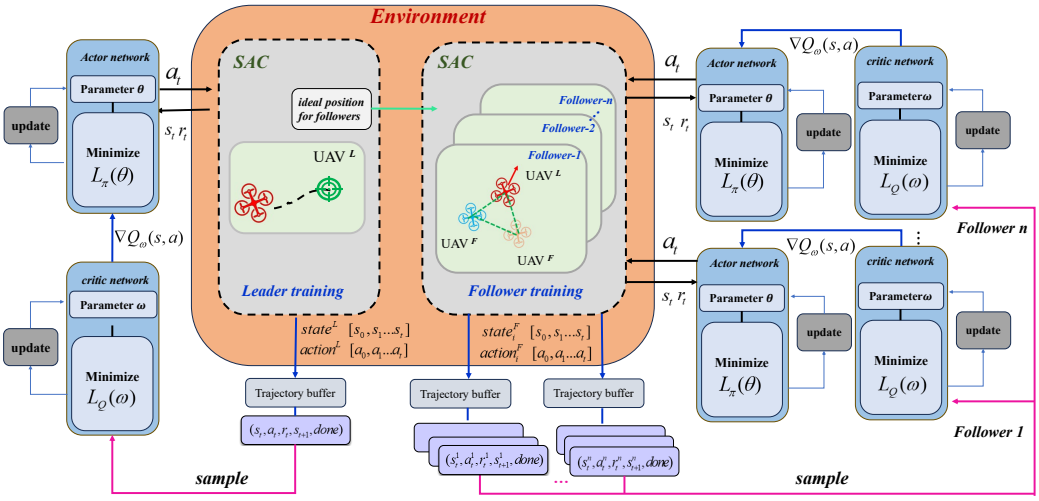


Figure 6. SAC training framework.

The SAC algorithm provides a data-driven realization of the leader and follower feedback policies. The leader policy is optimized for target-directed navigation, while the follower policies are optimized for formation regulation and local collision avoidance relative to the leader-generated reference. Theorem 1 establishes bounded formation deviation over finite obstacle-induced reconfiguration intervals and finite-time recovery under Assumption 3. Corollary 1 gives the corresponding geometric safety implication when sufficient nominal clearance is available. Target-reaching performance and unconditional collision avoidance are not implied by these theoretical results and are evaluated through the simulation experiments in Section 5.

5. Simulation Experiments and Result Analysis

5.1. Simulation Setup and Evaluation Protocol

To evaluate the proposed method, a Python 3.10.19-based multi-UAV cooperative simulation environment is constructed. Unless otherwise stated, the main comparative and ablation experiments use a three-UAV formation consisting of one leader and two followers. To further evaluate the scalability of the proposed framework with respect to formation size, additional experiments with one leader and three or four followers are conducted in Section 5.5. All UAVs adopt the same two-dimensional kinematic model and are equipped with forward-facing LiDAR sensors for local environmental perception. Static obstacles and configurable moving obstacles are used to construct scenarios with different passage geometries and dynamic disturbances. All experiments are conducted under Ubuntu 20.04 on a workstation equipped with an Intel Core i7-14700KF CPU (Intel Corporation, Santa Clara, CA, USA) and an NVIDIA RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

Within the simulator, UAV states and LiDAR measurements are updated at a common control interval, and formation-member states are exchanged synchronously. Policy outputs are mapped to the prescribed action ranges and directly applied to the discrete kinematic model.

For the proposed hierarchical framework, Stage 1 contains 500 episodes of leader-only training to obtain a stable navigation and collision-avoidance policy. After convergence, the leader parameters are frozen. Stage 2 then contains 3000 episodes of follower training, during which the leader trajectory provides the dynamic geometric reference for generating the followers' ideal formation positions. In both stages, the four map layouts are presented in the fixed cyclic order Map (a) $\rightarrow$ Map (b) $\rightarrow$ Map (c) $\rightarrow$ Map (d). For each map, ten predefined initial positions are used during training so that the policy is exposed to different initial navigation conditions. This is particularly relevant in dynamic environments, where different initial positions alter the encounter timing and relative geometry between the UAV formation and moving obstacles. The same initial-position set is used for all compared methods to maintain a consistent training distribution. The key SAC hyperparameters are listed in Table 2.

To reduce the influence of training randomness, all quantitative comparisons are repeated with three independent random seeds. For each seed, the complete training procedure is restarted from an independent initialization. Each trained policy is then evaluated for 10 episodes on each map using the same predefined initial-position set, yielding $N_{\text{total}} = 30$ evaluation episodes for every method–map combination. This protocol is intended to compare task success and formation-tracking performance under diverse but controlled narrow-passage conditions.

Task success rate and formation error are used as the primary evaluation metrics. An episode is regarded as successful only when all formation members complete the prescribed navigation task without collision before termination. The success rate is defined as

$$\kappa = \frac{N_{\text{success}}}{N_{\text{total}}} \times 100\%. \tag{38}$$

Here, N_{success} denotes the number of successful evaluation episodes, and N_{total} denotes the total number of evaluation episodes for the corresponding method–map combination.

Table 2. Hyperparameters of the SAC algorithm.

Parameter	Symbol	Value
Batch size	B	2048
Replay buffer capacity	D	100,000
Discount factor	γ	0.99
Soft update coefficient	τ	0.005
Initial entropy coefficient	α	0.20
Target entropy	$\mathcal{H}_{\text{target}}$	-2.0
Critic network learning rate	λ_Q	3×10^{-3}
Actor network learning rate	λ_π	1×10^{-3}
Entropy-coefficient learning rate	λ_α	1×10^{-4}
PER priority exponent	α_{PER}	0.6
PER importance-sampling exponent	β	$0.4 \rightarrow 1.0$
PER importance-sampling increment	$\Delta\beta$	5×10^{-4}
Optimizer	–	Adam

Note: The proposed SAC algorithm employs prioritized experience replay (PER) [35], which assigns sampling probabilities according to the priorities of experience samples. The priority exponent α_{PER} controls the degree of prioritized sampling, while the importance-sampling exponent β controls the correction for nonuniform sampling bias. After each replay update, β increases by $\Delta\beta$ until reaching 1.0.

In narrow environments, the ideal formation position of a follower may fall inside an obstacle, in which case the corresponding deviation does not accurately reflect formation-tracking capability. Therefore, formation error is evaluated only when the ideal formation position is geometrically feasible. In the following formulas, i indexes followers, j indexes evaluation episodes, and t indexes discrete time steps; $\mathbf{p}_{i,j}^{I}(t)$ and $\mathbf{p}_{i,j}^{F}(t)$ denote the ideal and actual positions of follower i, respectively, at time t in episode j. For the j-th successful episode, the valid sample set is

$$\mathcal{V}_j = \left\{ (i,t) \mid d_{\mathcal{O}_j(t)}\left(\mathbf{p}_{i,j}^{I}(t)\right) \geq d_{\text{safe}} \right\}, \tag{39}$$

where $\mathcal{O}_j(t)$ denotes the obstacle set at time t in episode j. If $|\mathcal{V}_j| > 0$, the episode-level formation error is defined as

$$\varepsilon_j = \frac{1}{|\mathcal{V}_j|} \sum_{(i,t)\in\mathcal{V}_j} \left\| \mathbf{p}_{i,j}^{F}(t) - \mathbf{p}_{i,j}^{I}(t) \right\|_2, \tag{40}$$

where $d_{\mathcal{O}_j(t)}(\mathbf{p})$ denotes the minimum distance from position $\mathbf{p}$ to the obstacle set at time t in episode j, and $d_{\text{safe}} = 0.5\,\text{m}$. Episodes with $|\mathcal{V}_j| = 0$ are excluded from the formation-error statistics while remaining included in the calculation of the task success rate. Let $\mathcal{J}_v$ denote the set of successful episodes satisfying $|\mathcal{V}_j| > 0$, and let $N_v = |\mathcal{J}_v|$. The reported formation error is $\bar{\varepsilon} \pm s_\varepsilon$, where

$$\bar{\varepsilon} = \frac{1}{N_v} \sum_{j\in\mathcal{J}_v} \varepsilon_j, \qquad s_\varepsilon = \sqrt{\frac{1}{N_v - 1} \sum_{j\in\mathcal{J}_v} (\varepsilon_j - \bar{\varepsilon})^2}. \tag{41}$$

A two-sided 95% confidence interval is also reported. Failed episodes are reflected by κ and are not included in the formation-error statistics. When $N_v = 0$, all formation-

error statistics are reported as N/A. When $N_v = 1$, only $\bar{\varepsilon}$ is reported, while the standard deviation and confidence interval are reported as N/A.

5.2. Training and Evaluation Environments

To evaluate the proposed method under different spatial constraints, four map layouts with different passage geometries are constructed together with corresponding dynamic-obstacle variants, as shown in Figures 7 and 8. In the static setting, Maps (a)–(c) contain central passage widths of d, $2d$, and $3d$, respectively, while Map (d) retains the passage width d but shifts the gap position to increase the diversity of spatial structures. Here, d denotes a reference passage width, and d, $2d$, and $3d$ specify the relative width ratios of the passages. The four maps participate cyclically in policy training.

In the dynamic setting, moving obstacles traveling back and forth along fixed trajectories with velocity v_d are superimposed on the corresponding static layouts. The UAVs therefore encounter both geometric passage constraints and time-varying local obstacle occupancy. The main UAV and environment parameters are summarized in Table 3.

All four map topologies are included in both policy training and quantitative evaluation. Accordingly, these experiments assess performance under the specified passage geometries and static/dynamic obstacle configurations, rather than generalization to map topologies held out from training.

Table 3. UAV and environment parameter settings.

Parameter	Symbol	Value
Speed	v	$[0, 4]$ m/s
Linear acceleration	a_c	$[-3.0, 3.0]$ m/s^2
Heading angle	ψ	$[-\pi, \pi]$ rad
Angular velocity	ω	$[-2, 2]$ rad/s
Sampling time	Δt	0.1 s
UAV equivalent body radius	r_{UAV}	0.5 m
Safe collision radius	d_{safe}	0.5 m
Obstacle-interaction threshold	d_{th}	1.5 m
LiDAR maximum range	$R_{\max}$	5.0 m
Formation spacing	l	4.0 m
Dynamic obstacle speed	v_d	2 m/s

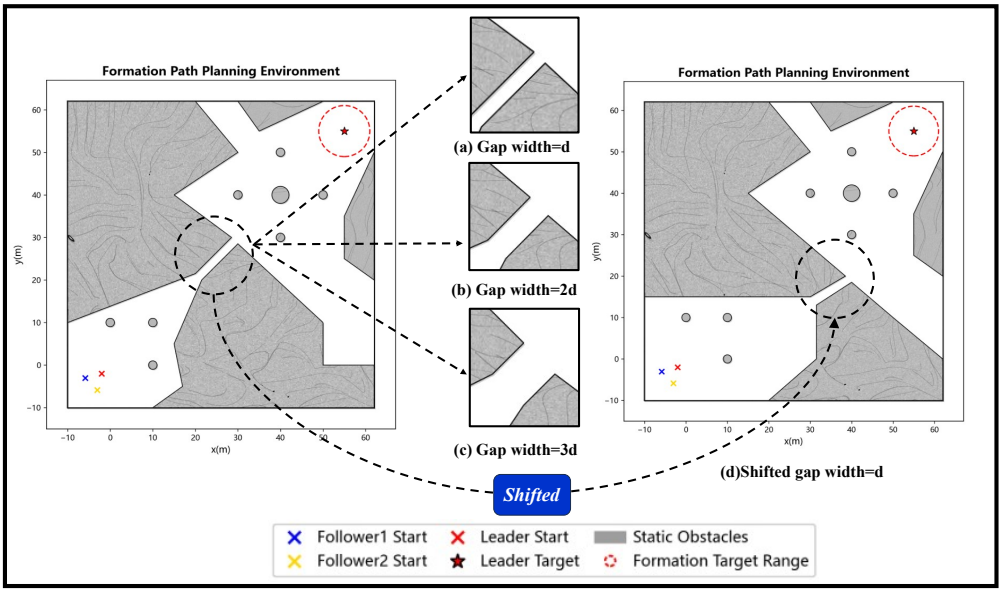


Figure 7. Static-obstacle environments.

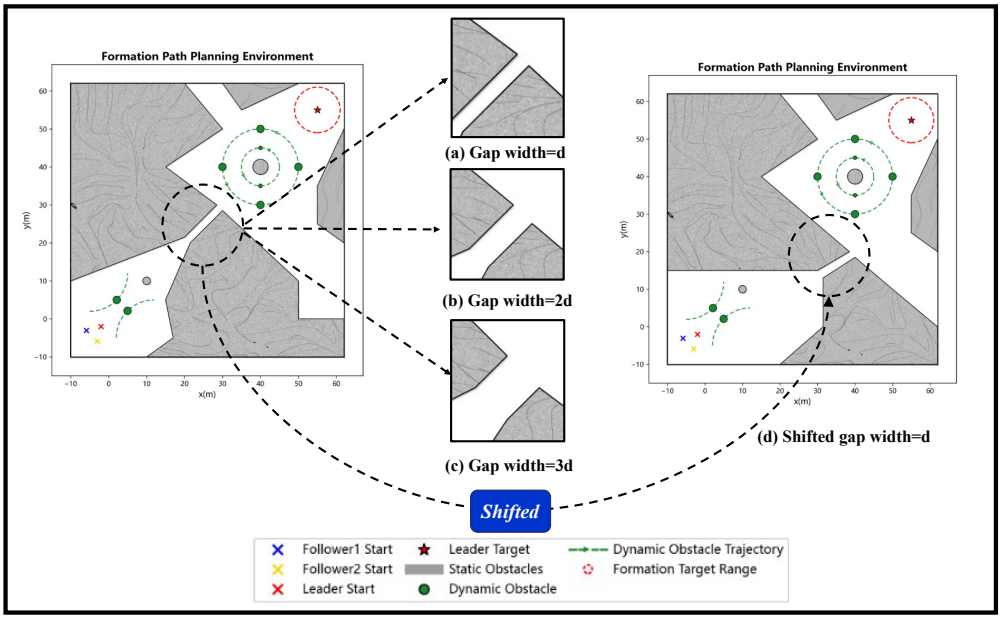


Figure 8. Dynamic-obstacle environments.

5.3. Qualitative Validation of Formation Reconfiguration

The formation trajectories in the static environments are shown in Figure 9. As the formation approaches a narrow region and the available free space decreases, the followers adapt their relative positions according to the local geometric constraints. In the narrowest Map (a), the lateral spacing is compressed and the formation passes through the gap in an approximately column-like configuration. In Map (b), only partial compression is required, whereas the original triangular geometry is largely retained in the wider Map (c). In Map (d), the leader adjusts its trajectory toward the shifted opening and the followers update their motion relative to the moving reference.

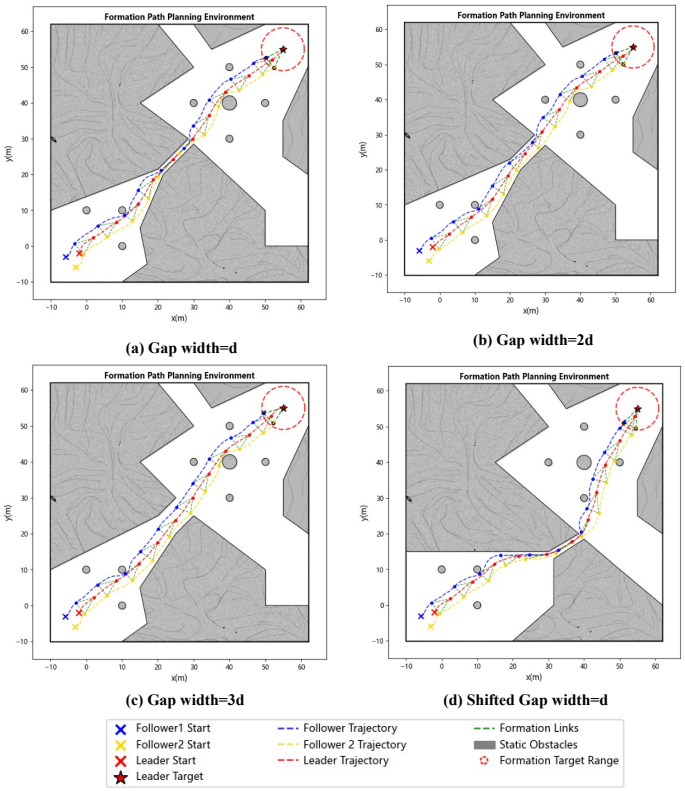


Figure 9. Formation trajectories in static scenarios.

The dynamic trajectories are shown in Figure 10. When a moving obstacle approaches a follower, the follower temporarily deviates from its ideal formation position to perform collision avoidance. After the obstacle leaves the interaction region, the follower gradually recovers toward the desired relative position. The formation therefore completes compression and recovery continuously according to local spatial constraints without predefined discrete formation modes or switching rules.

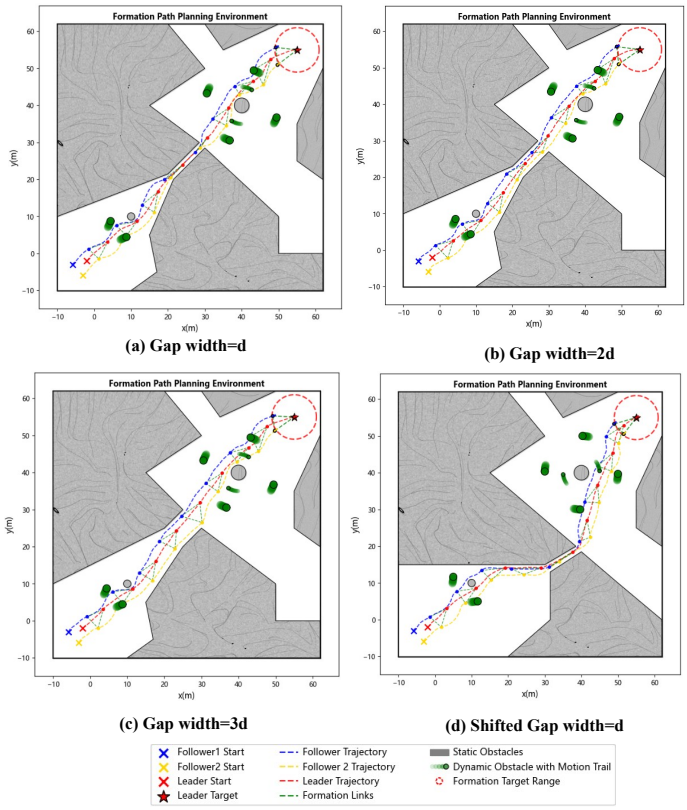


Figure 10. Formation trajectories in dynamic scenarios.

5.4. Comparative and Ablation Studies

To analyze the contribution of the main components of the proposed framework, four groups of experiments are conducted: heterogeneous-encoder ablation, reinforcement-learning algorithm comparison, formation-reward weighting ablation, and sensitivity analysis of the obstacle-interaction threshold d_{th}. All experiments use the common evaluation metrics and multi-seed protocol defined in Section 5.1.

5.4.1. Ablation Study of the Heterogeneous Encoder

To investigate the influence of state encoding on policy performance, five network combinations are compared under identical training conditions: GRU–CNN (G-C, proposed), LSTM–CNN (L-C), GRU–MLP (G-M), MLP–CNN (M-C), and MLP–MLP (M-M). GRU, Long Short-Term Memory (LSTM), and MLP are used to compare different encoders for proprioceptive states, whereas CNN and MLP are used to compare feature-extraction structures for LiDAR observations. The corresponding leader and follower training curves are shown in Figures 11 and 12.

The training curves indicate that all encoder variants can gradually establish navigation and formation-control policies in the static setting, while their differences become more evident after moving obstacles are introduced. G-C maintains a relatively stable convergence process for both the leader and followers in the dynamic setting.

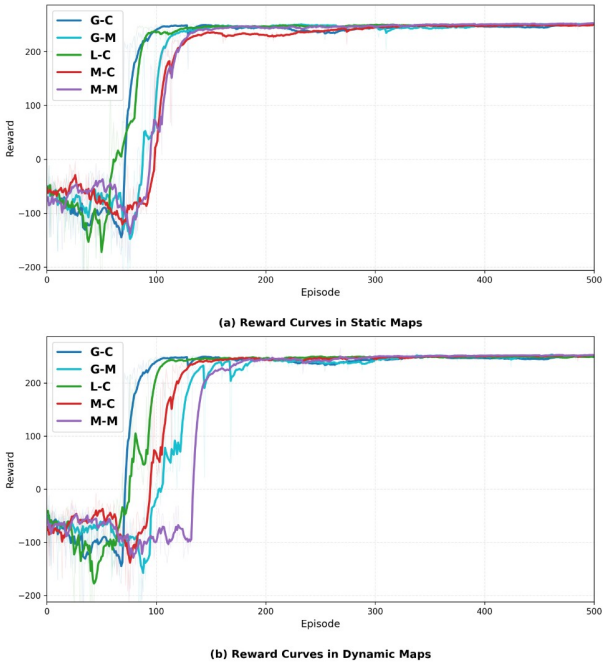


Figure 11. Leader reward convergence curves of different encoder structures in static and dynamic maps.

Table 4 summarizes the quantitative results. In the four static maps, all five encoder structures achieve 100% task success. G-C obtains the lowest formation errors in Maps (a)–(d), with values of 0.3176 m, 0.2734 m, 0.3002 m, and 0.4263 m, respectively, indicating improved formation-tracking accuracy under static geometric constraints.

After moving obstacles are introduced, the differences among the encoder structures become more pronounced. G-C maintains 100% task success in Maps (a)–(c) and 90% in Map (d), while achieving the lowest formation error in all four dynamic maps. Replacing the CNN branch with an MLP causes G-M to fail completely in dynamic Maps (a) and (d), and M-M also fails in these two maps.

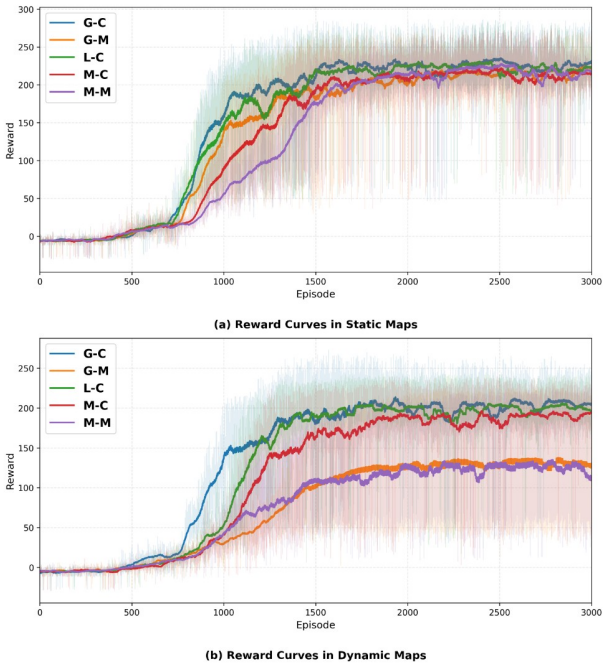


Figure 12. Follower reward convergence curves of different encoder structures in static and dynamic maps.

Table 4. Formation performance of different encoders in static and dynamic maps.

Map	Encoder	Static Maps		Dynamic Maps	
		Error ($\bar{\varepsilon}$, m)	Success (κ)	Error ($\bar{\varepsilon}$, m)	Success (κ)
Map (a)	GRU+CNN (Proposed)	**0.3176** ± 0.0173 [0.3110, 0.3242]	100%	**0.4536** ± 0.0850 [0.4219, 0.4853]	100%
	LSTM+CNN	0.3534 ± 0.0141 [0.3481, 0.3587]	100%	0.4835 ± 0.0719 [0.4567, 0.5103]	100%
	GRU+MLP	0.3427 ± 0.0109 [0.3386, 0.3467]	100%	N/A N/A	0.00%
	MLP+CNN	0.3721 ± 0.0132 [0.3671, 0.3770]	100%	0.5845 ± 0.0888 [0.5479, 0.6212]	83.33%
	MLP+MLP	0.3750 ± 0.0141 [0.3698, 0.3803]	100%	N/A N/A	0.00%
Map (b)	GRU+CNN (Proposed)	**0.2734** ± 0.0153 [0.2677, 0.2791]	100%	**0.4022** ± 0.0685 [0.3766, 0.4277]	100%
	LSTM+CNN	0.3042 ± 0.0125 [0.2995, 0.3089]	100%	0.4277 ± 0.0706 [0.4013, 0.4540]	100%
	GRU+MLP	0.3168 ± 0.0096 [0.3132, 0.3203]	100%	0.4091 ± 0.0615 [0.3837, 0.4345]	83.33%
	MLP+CNN	0.3419 ± 0.0105 [0.3380, 0.3459]	100%	0.5208 ± 0.0683 [0.4926, 0.5490]	83.33%
	MLP+MLP	0.3264 ± 0.0103 [0.3226, 0.3302]	100%	0.4939 ± 0.0463 [0.4748, 0.5130]	83.33%
Map (c)	GRU+CNN (Proposed)	**0.3002** ± 0.0164 [0.2940, 0.3063]	100%	**0.3838** ± 0.0427 [0.3679, 0.3998]	100%
	LSTM+CNN	0.3054 ± 0.0161 [0.2994, 0.3114]	100%	0.4029 ± 0.0485 [0.3848, 0.4211]	100%
	GRU+MLP	0.3124 ± 0.0126 [0.3077, 0.3171]	100%	0.3923 ± 0.0340 [0.3793, 0.4052]	100%
	MLP+CNN	0.3403 ± 0.0140 [0.3351, 0.3455]	100%	0.4806 ± 0.0587 [0.4587, 0.5025]	100%
	MLP+MLP	0.3499 ± 0.0123 [0.3453, 0.3545]	100%	0.4697 ± 0.0560 [0.4488, 0.4906]	100%
Map (d)	GRU+CNN (Proposed)	**0.4263** ± 0.0197 [0.4188, 0.4337]	100%	**0.5374** ± 0.1211 [0.4895, 0.5853]	90.00%
	LSTM+CNN	0.4288 ± 0.0247 [0.4193, 0.4382]	100%	0.5860 ± 0.1842 [0.5131, 0.6589]	90.00%
	GRU+MLP	0.4503 ± 0.0162 [0.4442, 0.4564]	100%	N/A N/A	0.00%
	MLP+CNN	0.4379 ± 0.0228 [0.4294, 0.4464]	100%	0.6223 ± 0.1409 [0.5628, 0.6818]	80.00%
	MLP+MLP	0.4969 ± 0.0181 [0.4901, 0.5036]	100%	N/A N/A	0.00%

Mean ± standard deviation; brackets denote 95% confidence intervals; bold denotes the lowest mean formation error per map/environment.

M-C achieves success rates of 83.33%, 83.33%, 100%, and 80% in Maps (a)–(d), respectively, with larger formation errors than G-C. L-C maintains the same success rates as G-C but produces larger formation errors in each dynamic map.

These results show that both the temporal evolution of proprioceptive states and the local spatial structure of LiDAR observations contribute to policy performance in dynamic environments. The GRU branch captures short-term variations in motion and formation states, while the CNN branch extracts local geometric features from LiDAR measurements. Their fusion provides the most balanced formation-tracking accuracy and task-completion performance among the evaluated encoder structures.

Notably, LSTM–CNN (L-C) achieves the same task success rates as GRU–CNN (G-C) in the dynamic environments but exhibits consistently higher formation-tracking errors. This difference may be related to the short temporal input and encoder complexity. With an observation window of $T_{\text{seq}} = 3$, the proprioceptive states mainly reflect short-term variations in UAV motion and formation deviations, for which the simpler GRU structure may already be adequate. In contrast, the additional cell state and gating operations of LSTM may offer limited benefit for such short sequences and the greater architectural complexity may make optimization more challenging under a fixed training budget. Consequently, the more complex LSTM encoder does not provide further performance gains in the present experimental setting.

5.4.2. Comparison with Reinforcement-Learning Baselines

Four reinforcement-learning configurations are compared. HSAC denotes the proposed hierarchical SAC framework. HPPO retains the same leader-pretraining/follower-training structure but replaces SAC with PPO. MASAC and MAPPO adopt centralized training and decentralized execution (CTDE), in which the leader and follower policies are updated jointly during training.

For a consistent comparison, all four methods employ the same G-C heterogeneous feature encoder, reducing the influence of differences in perceptual representation. Under CTDE, the centralized Q critic of MASAC estimates action values using the joint observation features and joint actions of all UAVs, whereas the centralized value critic of MAPPO estimates an individual state value for each agent using the joint observation features of all UAVs. For both MASAC and MAPPO, each agent maintains independently parameterized actor and critic networks, without parameter sharing among agents. During decentralized execution, each UAV's actor selects actions based only on its own local observations. MASAC and HSAC adopt the same G-C encoder architecture and SAC hyperparameters (Table 2). HPPO and MAPPO share the PPO settings listed in Table 5.

Table 5. Hyperparameters of the PPO-based baselines.

Parameter	Symbol	Value
Rollout length	N_{rollout}	2048
Minibatch size	B	256
Update epochs	K	10
Discount factor	γ	0.99
GAE coefficient	λ	0.95
Clipping ratio	ϵ	0.20
Actor learning rate	λ_π	2×10^{-4}
Critic learning rate	λ_V	1×10^{-3}
Entropy coefficient	c_{ent}	$0.005 \rightarrow 0.0005$
Initial log standard deviation	$\log \sigma_0$	-0.7
Optimizer	–	Adam

Table 5 summarizes the main training parameters of the PPO baselines. Each policy update uses a rollout of 2048 time steps, divided into minibatches of 256 samples and optimized for 10 epochs. The discount factor and GAE coefficient are set to 0.99 and 0.95, respectively, to balance long-horizon return estimation and the variance of advantage

estimates, while the clipping ratio of 0.20 limits the magnitude of each policy update. The actor and critic learning rates are set to 2×10^{-4} and 1×10^{-3}, respectively. The entropy coefficient is linearly reduced from 0.005 to 0.0005 over the first 100 PPO updates and then held constant, promoting policy exploration during the early stage of training while improving convergence stability in the later stage. The Gaussian policy is initialized with a log standard deviation of -0.7, and Adam is used to optimize the actor and critic networks. These parameters are kept identical for HPPO and MAPPO to ensure a fair comparison.

Different learning rates are used for PPO and SAC to reflect their distinct policy-update mechanisms. PPO employs on-policy optimization with multiple passes over each rollout and therefore uses smaller learning rates for stable updates. SAC uses experience replay and soft target-network updates, allowing relatively higher learning rates in the present configuration. Within each algorithm family, the compared methods use identical learning rates to ensure consistent settings.

In preliminary experiments, directly optimizing navigation, collision avoidance, and formation maintenance from the beginning prevented MASAC and MAPPO from reaching stable convergence within the prescribed training budget. A two-stage curriculum is therefore introduced. MASAC and MAPPO are each trained for a total of 3000 episodes, without a predefined allocation of episodes between the two curriculum stages. During Stage 1, the follower formation reward is disabled so that the three UAVs first learn navigation and collision avoidance. A 50-episode sliding window is used to calculate the proportion of episodes in which all three UAVs successfully reach their destinations. When this rate reaches 90% or higher, Stage 2 begins and the complete follower reward is activated. The leader and follower policies continue to be updated jointly in both stages. By contrast, HSAC and HPPO first complete leader training and then optimize the follower policies after the converged leader policy has been fixed. The corresponding training curves in the static and dynamic environments are shown in Figure 13.

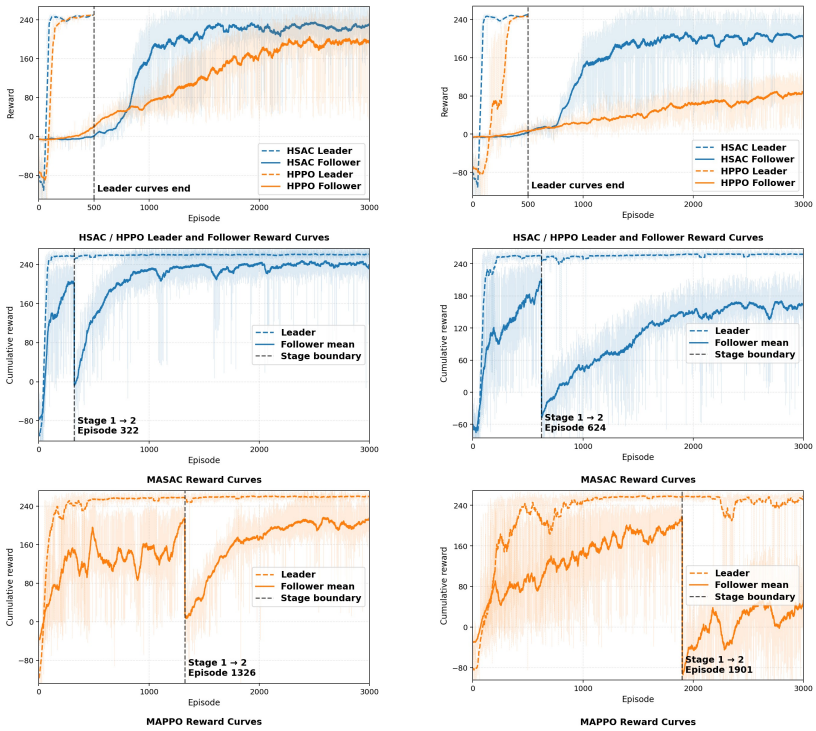


Figure 13. Training reward curves of HSAC, HPPO, MASAC, and MAPPO in static and dynamic maps: (**a**) static maps; (**b**) dynamic maps. For MASAC and MAPPO, the curriculum-stage boundary is marked at the episode where the 50-episode all-agent success rate first reaches 90% or higher.

Table 6 gives the quantitative comparison. In the static environments, all four methods achieve 100% task success. HSAC obtains the lowest formation errors in Maps (a)–(c), with values of 0.3176 m, 0.2734 m, and 0.3002 m, respectively. In Map (d), MASAC obtains a formation error of 0.4093 m, slightly lower than the 0.4263 m achieved by HSAC. These results indicate that all four methods can learn effective policies under relatively stable static geometric constraints.

Table 6. Formation performance of different algorithms in static and dynamic maps.

Map	Algorithm	Static Maps		Dynamic Maps	
		Error ($\bar{\varepsilon}$, m)	Success (κ)	Error ($\bar{\varepsilon}$, m)	Success (κ)
Map (a)	HSAC (Proposed)	**0.3176** ± 0.0173 [0.3110, 0.3242]	100%	**0.4536** ± 0.0850 [0.4219, 0.4853]	100%
	HPPO	0.4499 ± 0.0222 [0.4413, 0.4585]	100%	N/A N/A	0.00%
	MASAC	0.3630 ± 0.0101 [0.3592, 0.3668]	100%	0.6689 ± 0.0453 [0.6520, 0.6858]	100%
	MAPPO	0.4879 ± 0.0182 [0.4811, 0.4947]	100%	1.3854 ± 0.0629 [1.3541, 1.4167]	60.00%
Map (b)	HSAC (Proposed)	**0.2734** ± 0.0153 [0.2677, 0.2791]	100%	**0.4022** ± 0.0685 [0.3766, 0.4277]	100%
	HPPO	0.4329 ± 0.0197 [0.4256, 0.4403]	100%	N/A N/A	0.00%
	MASAC	0.2822 ± 0.0065 [0.2798, 0.2846]	100%	0.6093 ± 0.0492 [0.5910, 0.6277]	100%
	MAPPO	0.4509 ± 0.0153 [0.4451, 0.4566]	100%	1.3264 ± 0.0538 [1.2978, 1.3551]	53.33%
Map (c)	HSAC (Proposed)	**0.3002** ± 0.0164 [0.2940, 0.3063]	100%	**0.3838** ± 0.0427 [0.3679, 0.3998]	100%
	HPPO	0.4116 ± 0.0327 [0.3993, 0.4238]	100%	N/A N/A	0.00%
	MASAC	0.3053 ± 0.0074 [0.3026, 0.3081]	100%	0.5201 ± 0.0409 [0.5048, 0.5354]	100%
	MAPPO	0.4394 ± 0.0199 [0.4319, 0.4468]	100%	1.1835 ± 0.0382 [1.1631, 1.2038]	53.33%
Map (d)	HSAC (Proposed)	0.4263 ± 0.0197 [0.4188, 0.4337]	100%	**0.5374** ± 0.1211 [0.4895, 0.5853]	90.00%
	HPPO	0.6657 ± 0.0373 [0.6515, 0.6799]	100%	N/A N/A	0.00%
	MASAC	**0.4093** ± 0.0154 [0.4036, 0.4151]	100%	0.6368 ± 0.0435 [0.6192, 0.6544]	86.67%
	MAPPO	0.5172 ± 0.0175 [0.5107, 0.5238]	100%	1.8209 ± 0.1300 [1.7210, 1.9208]	30.00%

Mean ± standard deviation; brackets denote 95% confidence intervals; bold denotes the lowest mean formation error per map/environment.

The differences become more pronounced in the dynamic environments. HSAC achieves success rates of 100%, 100%, 100%, and 90% in Maps (a)–(d), with formation errors of 0.4536 m, 0.4022 m, 0.3838 m, and 0.5374 m, respectively. MASAC also maintains high task-completion performance, achieving success rates of 100%, 100%, 100%, and 86.67%, with corresponding formation errors of 0.6689 m, 0.6093 m, 0.5201 m, and 0.6368 m. Al-

though its success rates are close to those of HSAC, MASAC consistently produces larger formation errors in all four dynamic maps. MAPPO exhibits a more pronounced performance degradation, with success rates of 60%, 53.33%, 53.33%, and 30% in Maps (a)–(d), respectively, and substantially larger formation errors than HSAC. HPPO does not successfully complete the four dynamic-map evaluations under the current training configuration.

Moving obstacles strengthen the interaction among leader navigation, follower collision avoidance, and formation maintenance. Under joint training, the leader policy continues to change while the followers must adapt simultaneously to the evolving reference trajectory and dynamic obstacles. In the proposed hierarchical framework, follower learning is performed after the leader policy has converged and been fixed, providing a stable navigation reference for formation maintenance and local collision avoidance. The dynamic-map results therefore suggest that fixing a converged leader reference can improve formation-tracking consistency under moving-obstacle interference while preserving a high task-completion rate. From the comparison design, HSAC and HPPO use the same staged training structure, so their performance differences mainly reflect the combined effects of the SAC and PPO optimizers and their respective hyperparameter configurations. HSAC and MASAC both employ SAC with the same G-C encoder and optimization hyperparameters, but differ in staged training versus curriculum-assisted joint training under CTDE. The latter comparison also changes the critic information structure and thus cannot isolate the effect of a single factor. Overall, within the evaluated dynamic narrow-passage environments, the combination of staged task decomposition and SAC provides the most favorable balance between task completion and formation-tracking accuracy.

5.4.3. Ablation Study of Formation-Reward Weighting Strategies

To investigate the effect of the continuous adaptive formation-reward weight, the proposed Linear Decay strategy is compared with Fixed and Threshold-Switch strategies. In the Fixed strategy, the formation reward retains its full weight throughout the episode:

$$W_{\text{formation}}^{\text{Fixed}} = 1. \tag{42}$$

In the Threshold-Switch strategy, the formation-reward weight changes discretely according to the minimum obstacle distance:

$$W_{\text{formation}}^{\text{TS}} = \begin{cases} 0, & d_{\min}^{\text{obs}} \le d_{\text{sw}}, \\ 1, & d_{\min}^{\text{obs}} > d_{\text{sw}}, \end{cases} \tag{43}$$

where $d_{\text{sw}} = 1.0\,\text{m}$. The proposed Linear Decay strategy continuously adjusts the formation-reward weight according to Equation (36). The corresponding training curves are shown in Figure 14.

As shown in Table 7, all three strategies achieve 100% task success in the four static maps, while Linear Decay produces the lowest formation error in every layout. Fixed yields the largest formation error in all four static maps, whereas Threshold-Switch lies between the other two strategies. For example, in Map (d), the formation errors of Linear Decay, Threshold-Switch, and Fixed are 0.4263 m, 0.4460 m, and 0.5216 m, respectively.

With Fixed weighting, the full formation constraint remains active even when local collision avoidance requires a follower to deviate from its ideal formation position. The simultaneous action of the two objectives can increase trajectory correction and subsequent formation-recovery error. Threshold-Switch removes the formation constraint after the obstacle distance crosses the switching boundary, but its weight changes directly between 1 and 0. Linear Decay instead weakens the formation constraint gradually as the obstacle ap-

proaches and restores it as free space increases, providing a continuous transition between formation maintenance and collision avoidance.

Table 7. Formation performance under different reward weighting strategies in static and dynamic maps.

Map	Strategy	Static Maps		Dynamic Maps	
		Error ($\bar{\varepsilon}$, m)	Success (κ)	Error ($\bar{\varepsilon}$, m)	Success (κ)
Map (a)	Linear Decay (Proposed)	**0.3176** ± 0.0173 [0.3110, 0.3242]	100%	**0.4536** ± 0.0850 [0.4219, 0.4853]	100%
	Threshold-Switch	0.3351 ± 0.0137 [0.3300, 0.3402]	100%	0.6189 ± 0.0988 [0.5798, 0.6579]	90.00%
	Fixed	0.3919 ± 0.0175 [0.3853, 0.3984]	100%	0.5203 ± 0.0916 [0.4834, 0.5573]	86.67%
Map (b)	Linear Decay (Proposed)	**0.2734** ± 0.0153 [0.2677, 0.2791]	100%	**0.4022** ± 0.0685 [0.3766, 0.4277]	100%
	Threshold-Switch	0.2978 ± 0.0096 [0.2942, 0.3014]	100%	0.5799 ± 0.1048 [0.5408, 0.6190]	100%
	Fixed	0.3614 ± 0.0133 [0.3565, 0.3664]	100%	0.4670 ± 0.0931 [0.4316, 0.5023]	96.67%
Map (c)	Linear Decay (Proposed)	**0.3002** ± 0.0164 [0.2940, 0.3063]	100%	**0.3838** ± 0.0427 [0.3679, 0.3998]	100%
	Threshold-Switch	0.3273 ± 0.0157 [0.3215, 0.3332]	100%	0.5735 ± 0.0854 [0.5416, 0.6054]	100%
	Fixed	0.3850 ± 0.0199 [0.3776, 0.3925]	100%	0.4738 ± 0.0834 [0.4420, 0.5055]	96.67%
Map (d)	Linear Decay (Proposed)	**0.4263** ± 0.0197 [0.4188, 0.4337]	100%	**0.5374** ± 0.1211 [0.4895, 0.5853]	90.00%
	Threshold-Switch	0.4460 ± 0.0172 [0.4394, 0.4525]	100%	0.6684 ± 0.0906 [0.6318, 0.7049]	86.67%
	Fixed	0.5216 ± 0.0244 [0.5125, 0.5308]	100%	0.7190 ± 0.2201 [0.6095, 0.8285]	60.00%

Mean ± standard deviation; brackets denote 95% confidence intervals; bold denotes the lowest mean formation error per map/environment.

The differences become larger in the dynamic maps. Linear Decay achieves success rates of 100%, 100%, 100%, and 90% in Maps (a)–(d), respectively, and produces the lowest formation error in all four cases. Threshold-Switch obtains success rates of 90%, 100%, 100%, and 86.67%, whereas Fixed decreases to 86.67%, 96.67%, 96.67%, and 60%. In dynamic Map (d), the formation errors of Linear Decay, Threshold-Switch, and Fixed are 0.5374 m, 0.6684 m, and 0.7190 m, respectively. These results demonstrate that continuous adjustment of the formation-reward weight provides a smoother balance between formation maintenance and collision avoidance, improving both formation tracking and task completion under dynamic constraints.

Because the three strategies modify the magnitude and activation pattern of the formation-reward term, their cumulative training returns are not on an identical numerical scale. Figure 14 is therefore used to compare the training process and convergence trend, while task success rate and formation error in Table 7 provide the common performance measures.

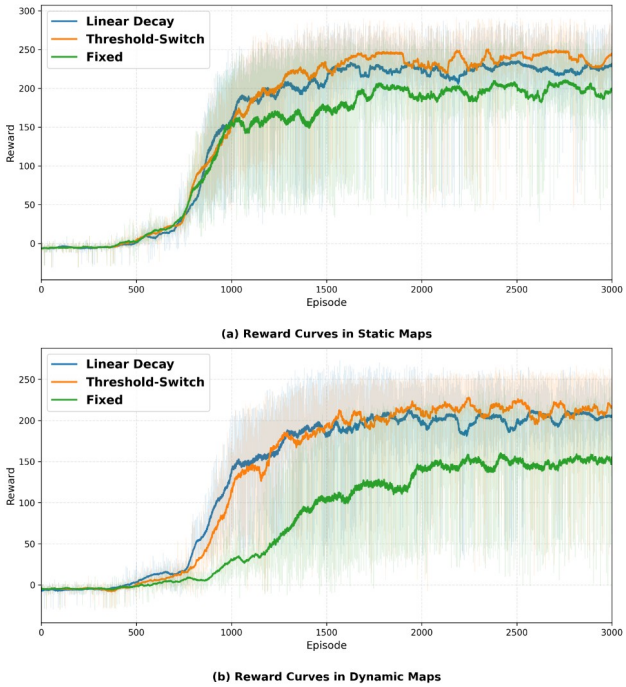


Figure 14. Reward convergence curves under different formation-reward weighting strategies in static and dynamic maps.

5.4.4. Sensitivity Analysis of the Obstacle-Interaction Threshold

To analyze the sensitivity of the proposed framework to the shared obstacle-interaction threshold d_{th}, experiments are conducted with

$$d_{\text{th}} \in \{1.0, 1.5, 3.0, 5.0\}\ \text{m}. \tag{44}$$

The corresponding sensitivity curves are shown in Figure 15.

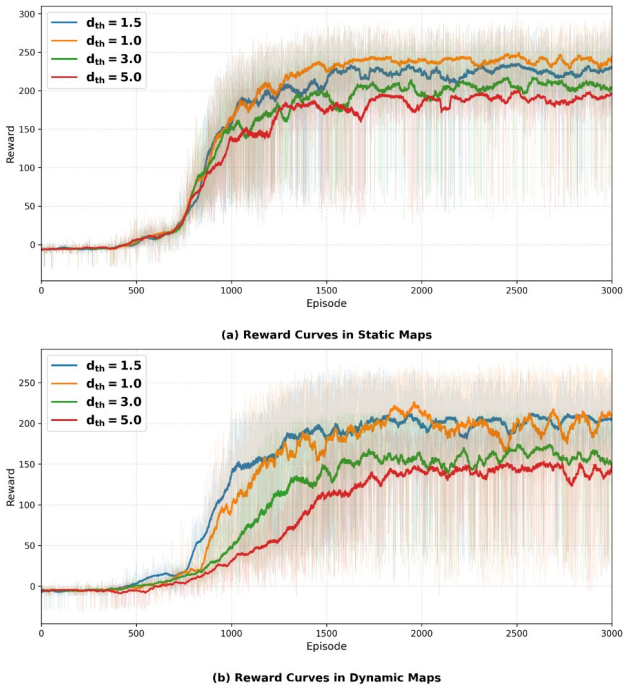


Figure 15. Reward convergence curves under different obstacle-interaction thresholds d_{th} in static and dynamic maps.

The safety distance is fixed at $d_{\text{safe}} = 0.5\,\text{m}$, while the remaining reward parameters, network structure, training episodes, map order, and random-seed settings are kept unchanged. The main experiments use $d_{\text{th}} = 1.5\,\text{m}$.

The threshold d_{th} determines both the spatial range of the obstacle-avoidance reward and the interval over which the formation-reward weight decays from 1 to 0. A smaller d_{th} retains a stronger formation constraint until the UAV is closer to an obstacle, which favors smaller formation error but shortens the available distance for anticipatory avoidance. A larger d_{th} activates obstacle response earlier but weakens the formation constraint over a wider spatial region.

Table 8 shows that d_{th} directly affects formation accuracy and dynamic obstacle-avoidance performance. When $d_{\text{th}} = 1.0\,\text{m}$, the formation errors in the four static maps are 0.2825 m, 0.2422 m, 0.2492 m, and 0.3960 m, respectively, which are the lowest values among the tested thresholds. The formation errors also remain low in the dynamic environments. However, the success rate decreases to 76.67% in dynamic Map (a) and 43.33% in Map (d), indicating that the shorter response distance can limit collision-avoidance capability in more challenging dynamic conditions.

As d_{th} increases to 3.0 and 5.0 m, the formation-reward weight begins to decay farther from obstacles, and the formation error shows an overall increasing trend in the static maps.

Table 8. Formation performance under different d_{th} values in static and dynamic maps.

Map	d_{th}	Static Maps		Dynamic Maps	
		Error ($\bar{\varepsilon}$, m)	Success (κ)	Error ($\bar{\varepsilon}$, m)	Success (κ)
Map (a)	1.0	**0.2825** ± 0.0182 [0.2757, 0.2893]	100%	**0.3770** ± 0.0912 [0.3376, 0.4164]	76.67%
	1.5 (Proposed)	0.3176 ± 0.0173 [0.3110, 0.3242]	100%	0.4536 ± 0.0850 [0.4219, 0.4853]	100%
	3.0	0.3527 ± 0.0160 [0.3467, 0.3586]	100%	0.5499 ± 0.0800 [0.5144, 0.5854]	73.33%
	5.0	0.4000 ± 0.0302 [0.3886, 0.4115]	100%	0.6755 ± 0.1310 [0.6237, 0.7273]	90.00%
Map (b)	1.0	**0.2422** ± 0.0167 [0.2360, 0.2485]	100%	**0.3085** ± 0.0792 [0.2778, 0.3392]	93.33%
	1.5 (Proposed)	0.2734 ± 0.0153 [0.2677, 0.2791]	100%	0.4022 ± 0.0685 [0.3766, 0.4277]	100%
	3.0	0.3231 ± 0.0110 [0.3190, 0.3272]	100%	0.5285 ± 0.1402 [0.4730, 0.5840]	90.00%
	5.0	0.3860 ± 0.0293 [0.3749, 0.3972]	100%	0.5351 ± 0.0562 [0.5133, 0.5569]	93.33%
Map (c)	1.0	**0.2492** ± 0.0181 [0.2424, 0.2559]	100%	**0.2858** ± 0.0719 [0.2585, 0.3132]	96.67%
	1.5 (Proposed)	0.3002 ± 0.0164 [0.2940, 0.3063]	100%	0.3838 ± 0.0427 [0.3679, 0.3998]	100%
	3.0	0.3717 ± 0.0098 [0.3681, 0.3754]	100%	0.5543 ± 0.1343 [0.5022, 0.6064]	93.33%
	5.0	0.4506 ± 0.0564 [0.4291, 0.4720]	100%	0.5011 ± 0.0450 [0.4840, 0.5182]	96.67%

Table 8. *Cont.*

Map	d_{th}	Static Maps		Dynamic Maps	
		Error ($\bar{\varepsilon}$, m)	Success (κ)	Error ($\bar{\varepsilon}$, m)	Success (κ)
Map (d)	1.0	**0.3960** ± 0.0224 [0.3874, 0.4045]	96.67%	**0.4796** ± 0.0829 [0.4295, 0.5297]	43.33%
	1.5 (Proposed)	0.4263 ± 0.0197 [0.4188, 0.4337]	100%	0.5374 ± 0.1211 [0.4895, 0.5853]	90.00%
	3.0	0.4833 ± 0.0287 [0.4722, 0.4944]	93.33%	0.5553 ± 0.0716 [0.5264, 0.5842]	86.67%
	5.0	0.5047 ± 0.0245 [0.4954, 0.5140]	96.67%	0.5958 ± 0.0664 [0.5684, 0.6232]	83.33%

Mean ± standard deviation; brackets denote 95% confidence intervals; bold denotes the lowest mean formation error per map/environment.

In the dynamic environments, the larger threshold expands the obstacle-response region but does not consistently improve task success and generally produces larger formation errors.

With $d_{\text{th}} = 1.5\,\text{m}$, the success rates in dynamic Maps (a)–(c) reach 100%, while Map (d) achieves 90%. The corresponding formation errors are 0.4536 m, 0.4022 m, 0.3838 m, and 0.5374 m, respectively. These results show that a small threshold favors compact formation tracking but can delay collision avoidance, whereas a large threshold releases the formation constraint earlier and increases formation deviation. Among the tested values, $d_{\text{th}} = 1.5\,\text{m}$ provides a favorable balance between task success and formation-tracking accuracy.

5.5. Scalability Evaluation with Larger Formations

To further evaluate the scalability of the proposed framework with respect to formation size, additional experiments were conducted by increasing the number of followers from two to three and four, resulting in four-UAV and five-UAV formations, respectively. The original hierarchical leader–follower training framework, G-C network architecture, and adaptive reward mechanism were retained, while additional desired formation positions were assigned to the newly introduced followers according to the formation model described in Section 3.2. Both static and dynamic narrow-passage environments were considered.

Figure 16 presents the reward convergence curves for the enlarged formations. All four configurations exhibit clear learning and stabilization trends within the prescribed training process. Compared with the three-follower case, the four-follower formation generally shows a more gradual convergence process, particularly in the dynamic environment. This behavior is expected because increasing the number of followers introduces additional inter-UAV interactions and collision-avoidance constraints, thereby increasing the coordination difficulty during training. Nevertheless, stable reward trends can still be established for both enlarged formation sizes.

To assess the effect of increasing formation size on task performance, the HSAC results for the original two-follower configuration in Table 6 (Section 5.4.2) are compared with the three- and four-follower results in Table 9. Overall, as the number of followers increases from two to three and four, task success rates tend to decrease and formation-tracking errors increase. For example, in static Map (d), the success rates for two, three, and four followers are 100.00%, 83.33%, and 80.00%, respectively, with corresponding formation-tracking errors of 0.4263 m, 0.5012 m, and 0.5526 m. In dynamic Map (d), the success rates are 90.00%, 80.00%, and 73.33%, respectively, while the corresponding errors increase

from 0.5374 m to 0.5834 m and 0.7648 m. These results indicate that task completion and formation maintenance become more challenging as the formation size increases, with the increase in formation-tracking error being particularly pronounced for the four-follower configuration in dynamic environments. Nevertheless, the proposed method retains a certain level of task-completion capability in the enlarged configurations, demonstrating scalability within the investigated formation sizes.

Representative trajectories are shown in Figure 17. In both the three-follower and four-follower cases, the formation can adjust its relative geometry when approaching the narrow passage, compress around the constrained region, and subsequently recover while progressing toward the target. Similar adaptive behavior is maintained when moving obstacles are introduced, where individual followers temporarily deviate from their desired formation positions to satisfy local collision-avoidance requirements. The trajectories therefore indicate that the proposed formation adaptation mechanism remains effective when additional followers are introduced.

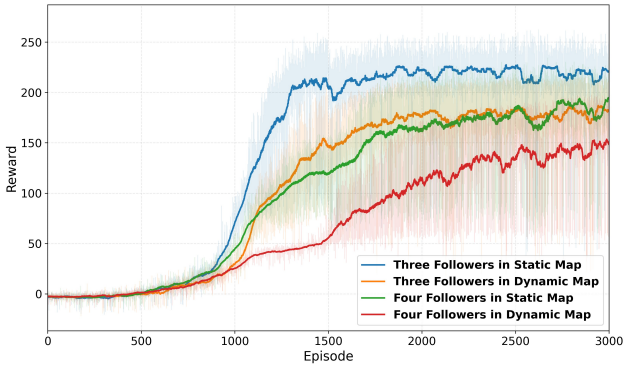
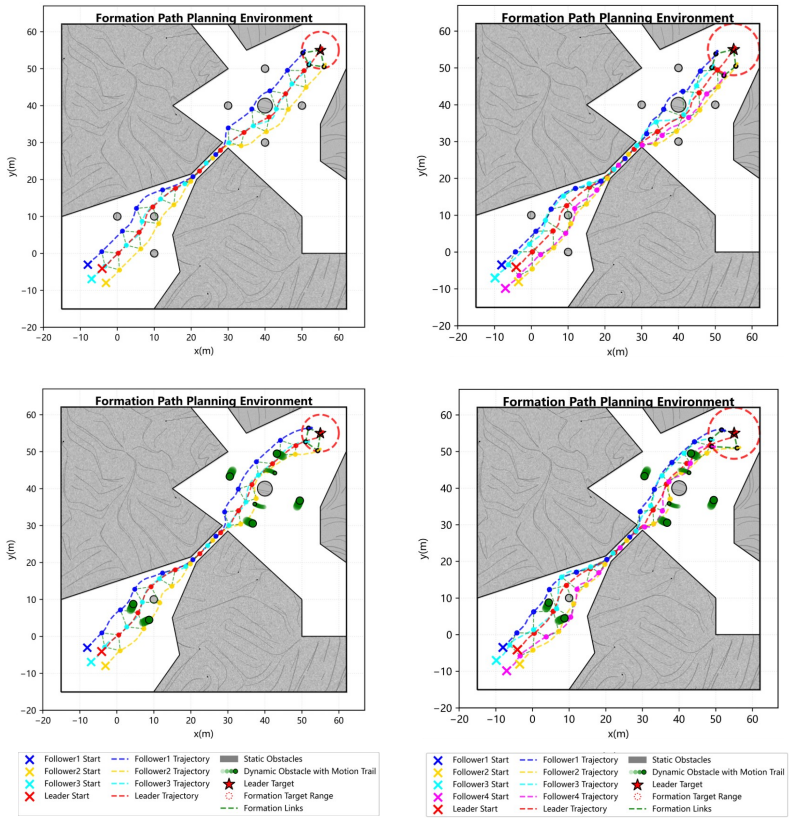


Figure 16. Reward convergence curves for enlarged formations with three and four followers in static and dynamic environments.

Table 9. Formation performance with different numbers of followers in static and dynamic maps.

Map	Number of Followers	Static Maps		Dynamic Maps	
		Error ($\bar{\varepsilon}$, m)	Success (κ)	Error ($\bar{\varepsilon}$, m)	Success (κ)
Map (a)	3	0.3682 ± 0.0200 [0.3604, 0.3760]	93.33%	0.4979 ± 0.0895 [0.4625, 0.5333]	90.00%
	4	0.4017 ± 0.0265 [0.3910, 0.4124]	86.67%	0.6510 ± 0.0966 [0.6111, 0.6909]	83.33%
Map (b)	3	0.3336 ± 0.0166 [0.3274, 0.3398]	100.00%	0.4397 ± 0.0707 [0.4123, 0.4671]	93.33%
	4	0.3906 ± 0.0248 [0.3808, 0.4004]	90.00%	0.6651 ± 0.0919 [0.6280, 0.7022]	86.67%
Map (c)	3	0.3647 ± 0.0190 [0.3576, 0.3718]	100.00%	0.3918 ± 0.0575 [0.3695, 0.4141]	93.33%
	4	0.3849 ± 0.0199 [0.3770, 0.3928]	90.00%	0.6322 ± 0.0879 [0.5967, 0.6677]	86.67%
Map (d)	3	0.5012 ± 0.0249 [0.4909, 0.5115]	83.33%	0.5834 ± 0.1217 [0.5320, 0.6348]	80.00%
	4	0.5526 ± 0.0346 [0.5380, 0.5672]	80.00%	0.7648 ± 0.1473 [0.6995, 0.8301]	73.33%

Results are reported as mean ± standard deviation; values in brackets denote the 95% confidence interval.



(a) Three Followers Trajectory **(b) Four Followers Trajectory**

Figure 17. Representative trajectories of enlarged formations in static (top row) and dynamic (bottom row) environments: (**a**) one leader with three followers; (**b**) one leader with four followers.

These results demonstrate that the proposed framework is not restricted to the original one-leader–two-follower configuration and can be extended to moderately larger formations without redesigning the basic hierarchical training architecture. At the same time, the slower convergence observed with increasing formation size suggests that the coordination and policy-learning tasks become more challenging as more UAVs are introduced. The quantitative results likewise show reduced task success rates and increased formation-tracking errors for the four-follower formation, especially in dynamic environments. Therefore, the present experiments support scalability within the investigated formation sizes, while scalability to substantially larger swarms remains to be further evaluated.

6. Discussion and Limitations

The expanded experiments provide complementary evidence for the main components of the proposed framework and its scalability. The algorithm comparison shows that the task-attribute-based HSAC design is particularly robust when dynamic obstacles strengthen the coupling among leader navigation, follower collision avoidance, and formation tracking. The HPPO result further indicates that the benefit cannot be attributed to hierarchy alone under the present training budget, while the curriculum-assisted MASAC and MAPPO comparisons indicate that fixing a converged leader reference can improve formation-tracking consistency under dynamic interference. The encoder ablation supports the use of separate recurrent and convolutional branches for temporally correlated proprioceptive variables and spatially structured LiDAR measurements. The reward-weighting ablation shows that continuous relaxation provides a more favorable balance than fixed or threshold-switched formation constraints, and the d_{th} sensitivity study demonstrates an explicit trade-off between formation accuracy and obstacle-response margin. The scalability experiments

further show that the same hierarchical framework can accommodate formations with three and four followers without structural modification, while the more gradual convergence observed with increasing formation size suggests increased coordination difficulty as more UAVs are introduced.

These conclusions remain bounded by the simulation assumptions. The experiments use a two-dimensional kinematic model with accurate state observations and reliable local information exchange. They therefore demonstrate the effectiveness of the hierarchical training strategy, heterogeneous state encoding, and adaptive formation regulation within the investigated simulation domain rather than establishing equivalent performance in real multi-UAV systems. Similarly, the bounded-deviation and finite-time recovery properties established in Section 3.3 are conditional on finite reconfiguration intervals and the local formation-recovery assumption, while the associated safety implication additionally requires sufficient geometric margins. They do not constitute unconditional guarantees under arbitrary perception degradation, communication interruption, or unknown disturbances. Because all four geometric layouts used for evaluation also participate in the fixed cyclic training set, the present experiments assess robustness across heterogeneous trained environments rather than generalization to completely unseen topologies; evaluation on held-out map layouts remains an important direction for future work. In addition, the scalability evaluation is currently limited to formations containing up to four followers; the behavior of the framework in substantially larger swarms, where inter-agent interactions and computational requirements become more significant, has not yet been systematically investigated.

In real narrow environments, structural occlusion, multipath propagation, and electromagnetic interference may degrade satellite-based positioning or lead to GNSS-denied operation. UAVs must then rely on inertial sensing, vision- or LiDAR-based localization, and multisensor fusion, for which localization drift, attitude-estimation errors, and temporal misalignment may affect the calculation of target bearings and formation deviations. The LiDAR observations used in the present simulation are also idealized and do not include ranging noise, occlusion, missed detections, or motion distortion. Their finite field of view and sensing range further limit direct perception of rear blind spots and rapidly approaching threats outside the sensing region. In addition, the current simulation assumes synchronous, low-latency, and lossless information exchange among formation members. In practice, communication delay, bandwidth limitations, packet loss, and asynchronous updates may cause followers to construct ideal formation positions from outdated leader states, thereby reducing formation accuracy and narrowing the available safety margin in confined spaces. Although the GRU branch can exploit short-term historical information to represent state trends, its effectiveness under substantial sensing uncertainty or persistent communication degradation remains to be evaluated.

The current model describes constant-altitude motion in a two-dimensional plane using linear acceleration and angular velocity as high-level control inputs. It does not fully capture roll, pitch, vertical motion, aerodynamic coupling, wind disturbances, actuator lag, or tracking errors of the low-level flight controller. The dynamic obstacles considered in the experiments also follow relatively regular motion patterns and do not cover the full range of high-speed, variable-speed, or nonperiodic threats. Moreover, the leader–follower structure depends on the continued availability of the leader state; leader sensing degradation, actuator failure, or communication loss may therefore disrupt the geometric reference used by the followers, while online leader reassignment and formation reconfiguration are not yet included. The G-C encoder and policy networks have thus far been evaluated on a desktop computing platform, and their inference latency, memory requirements,

and sustainable control frequency on resource-constrained onboard computers require further assessment.

Future work will progressively increase the fidelity of the validation environment. Localization errors, sensor noise, observation loss, communication delay and packet loss, wind disturbances, and model-parameter perturbations will first be introduced into the existing simulation to quantify performance degradation under different uncertainty levels. The hierarchical coordination framework will also be evaluated with larger-scale swarms and extended to heterogeneous UAV teams with different dynamics, sensing, and actuation capabilities. The framework will then be extended to high-fidelity three-dimensional models that include actuator dynamics, more complex moving threats, and broader flight-envelope constraints, together with leader reconfiguration and safe-degradation mechanisms under component or communication failures. Finally, flight-controller-in-the-loop, hardware-in-the-loop, and onboard-computing tests will be conducted before multi-UAV flight experiments, thereby further clarifying the applicability and engineering potential of the proposed method.

7. Conclusions

This paper presented a hierarchical leader–follower reinforcement-learning framework for adaptive multi-UAV formation navigation in narrow-passage environments. The cooperative task is organized according to the task attributes of the leader and followers: the leader is first trained independently to establish a stable navigation reference, after which its policy is frozen and the followers learn formation regulation and local collision avoidance relative to that reference while retaining target-directed progression. A GRU–CNN heterogeneous dual-stream encoder extracts temporally correlated proprioceptive features and spatially structured LiDAR features, while a distance-dependent continuous formation-weighting mechanism coordinates the competing objectives of formation preservation and safe passage.

Comprehensive comparative and ablation experiments clarify the contribution of each component. In the algorithm comparison, HSAC maintains 90–100% success across the four dynamic layouts and consistently smaller formation errors than MASAC and MAPPO. In particular, although curriculum-assisted MASAC achieves task-success rates comparable to HSAC, HSAC retains substantially lower formation errors, supporting the benefit of fixing a converged leader reference for formation tracking under dynamic interference. The comparison with HPPO further highlights the contribution of the SAC optimizer under the same hierarchical structure. Encoder ablation shows that the G-C structure provides the most consistent combination of formation accuracy and task success across static and dynamic environments. Reward-weighting ablation further shows that Linear Decay outperforms both Fixed and Threshold-Switch strategies in formation error and provides more reliable task completion under dynamic constraints. Finally, the d_{th} sensitivity experiments reveal an explicit trade-off: a small threshold preserves tighter formations but can delay obstacle response, whereas a large threshold relaxes the formation prematurely; $d_{\text{th}} = 1.5\,\text{m}$ provides a favorable balance in the investigated environments. Additional scalability experiments with three and four followers demonstrate that the proposed hierarchical framework can accommodate larger formations without modifying the basic training architecture, while maintaining effective policy learning and adaptive narrow-passage navigation. However, as the number of followers increases, task success rates decrease and formation-tracking errors increase, particularly in dynamic environments. These results indicate that the proposed method exhibits scalability within the investigated formation sizes, although its performance degrades as the formation size increases. Together with the theoretical analysis of bounded formation deviation, finite-time recovery, and the associ-

ated geometric safety implication, these results support the effectiveness of the proposed framework within the stated modeling assumptions.

The reported navigation and formation performance is demonstrated on the four passage topologies included in training; generalization to held-out unseen layouts has not yet been evaluated. The present study remains limited to two-dimensional constant-altitude kinematic simulations with idealized sensing and communication conditions and without physical flight validation. Future work will investigate robustness to localization errors, sensor noise, communication delay and packet loss, wind disturbances, and more complex dynamic threats; extend the framework to three-dimensional dynamics and fault-tolerant leader reconfiguration; and conduct onboard-computing, hardware-in-the-loop, and real multi-UAV flight experiments.

Author Contributions: Conceptualization, J.L., H.Y., D.W. and L.W.; methodology, J.L., H.Y., D.W. and L.W.; software, J.L.; validation, H.Y., D.W. and L.W.; formal analysis, H.Y., D.W. and L.W.; investigation, J.L., H.Y., D.W. and L.W.; data curation, H.Y.; writing—original draft preparation, J.L.; writing—review and editing, H.Y.; visualization, J.L. and H.Y.; supervision, H.Y., D.W. and L.W.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number 62003130 and the Fundamental Research Funds for the Central Universities grant number B230201055.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to prohibitive costs associated with data preparation and hosting within the terms of this research project.

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

References

1. Do, H.T.; Hua, H.T.; Nguyen, M.T.; Nguyen, C.V.; Nguyen, H.T.T.; Nguyen, H.T.; Nguyen, N.T.T. Formation Control Algorithms for Multiple-UAVs: A Comprehensive Survey. *EAI Endorsed Trans. Ind. Netw. Intell. Syst.* **2021**, *8*, e3. https://doi.org/10.4108/eai.10-6-2021.170230.
2. Yang, Y.; Xiong, X.; Yan, Y. UAV formation trajectory planning algorithms: A review. *Drones* **2023**, *7*, 62. https://doi.org/10.3390/drones7010062.
3. Bu, Y.; Yan, Y.; Yang, Y. Advancement challenges in UAV swarm formation control: A comprehensive review. *Drones* **2024**, *8*, 320. https://doi.org/10.3390/drones8070320.
4. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. https://doi.org/10.1109/ACCESS.2019.2909530.
5. Consolini, L.; Morbidi, F.; Prattichizzo, D.; Tosques, M. Leader–follower formation control of nonholonomic mobile robots with input constraints. *Automatica* **2008**, *44*, 1343–1349. https://doi.org/10.1016/j.automatica.2007.09.019.
6. Benzerrouk, A.; Adouane, L.; Martinet, P. Stable navigation in formation for a multi-robot system based on a constrained virtual structure. *Robot. Auton. Syst.* **2014**, *62*, 1806–1815. https://doi.org/10.1016/j.robot.2014.07.004.
7. Balch, T.; Arkin, R.C. Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **1998**, *14*, 926–939. https://doi.org/10.1109/70.736776.
8. Liu, H.; Chen, J.; Feng, J.; Zhao, H. An improved RRT* UAV formation path planning algorithm based on goal bias and node rejection strategy. *Unmanned Syst.* **2023**, *11*, 317–326. https://doi.org/10.1142/S2301385023500140.
9. Fang, Y.; Yao, Y.; Zhu, F.; Chen, K. Piecewise-potential-field-based path planning method for fixed-wing UAV formation. *Sci. Rep.* **2023**, *13*, 2234. https://doi.org/10.1038/s41598-023-28087-0.
10. Han, L.; Wu, X.; Sun, X. Hybrid path planning algorithm for mobile robot based on A* algorithm fused with DWA. In *Proceedings of the 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 26–28 May 2023*; IEEE: New York, NY, USA, 2023; Volume 3, pp. 1465–1469. https://doi.org/10.1109/ICIBA56860.2023.10165386.

11. Ouyang, Q.; Wu, Z.; Cong, Y.; Wang, Z. Formation control of unmanned aerial vehicle swarms: A comprehensive review. *Asian J. Control* **2023**, *25*, 570–593. https://doi.org/10.1002/asjc.2806.
12. Bai, Y.; Zhao, H.; Zhang, X.; Chang, Z.; Jäntti, R.; Yang, K. Toward autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 3038–3067. https://doi.org/10.1109/COMST.2023.3323344.
13. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Studies in Systems, Decision and Control; Vamvoudakis, K.G., Wan, Y., Lewis, F.L., Cansever, D., Eds.; Springer: Cham, Switzerland, 2021; Volume 325, pp. 321–384. https://doi.org/10.1007/978-3-030-60990-0_12.
14. Tang, J.; Liang, Y.; Li, K. Dynamic scene path planning of UAVs based on deep reinforcement learning. *Drones* **2024**, *8*, 60. https://doi.org/10.3390/drones8020060.
15. Liu, Z.; Li, J.; Shen, J.; Wang, X.; Chen, P. Leader–follower UAVs formation control based on a deep Q-network collaborative framework. *Sci. Rep.* **2024**, *14*, 4674. https://doi.org/10.1038/s41598-024-54531-w.
16. 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: A task-specific curriculum-based MADRL approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 10894–10908. https://doi.org/10.1109/TNNLS.2023.3245124.
17. Qie, H.; Shi, D.; Shen, T.; Xu, X.; Li, Y.; Wang, L. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access* **2019**, *7*, 146264–146272. https://doi.org/10.1109/ACCESS.2019.2943253.
18. Zhao, W.; Chu, H.; Miao, X.; Guo, L.; Shen, H.; Zhu, C.; Zhang, F.; Liang, D. Research on the Multiagent Joint Proximal Policy Optimization Algorithm Controlling Cooperative Fixed-Wing UAV Obstacle Avoidance. *Sensors* **2020**, *20*, 4546. https://doi.org/10.3390/s20164546.
19. Fang, C.; Yang, F.; Pan, Q. Multi-UAV Collaborative Path Planning Based on Multi-Agent Soft Actor Critic. *Sci. Sin. Inform.* **2024**, *54*, 1871–1883. https://doi.org/10.1360/SSI-2024-0050.
20. Wang, Y.; Cui, Y.; Yang, Y.; Li, Z.; Cui, X. Multi-UAV Path Planning for Air-Ground Relay Communication Based on Mix-Greedy MAPPO Algorithm. *Drones* **2024**, *8*, 706. https://doi.org/10.3390/drones8120706.
21. Zhou, Y.; Shu, J.; Zheng, X.; Hao, H.; Song, H. Real-time route planning of unmanned aerial vehicles based on improved soft actor-critic algorithm. *Front. Neurorobot.* **2022**, *16*, 1025817. https://doi.org/10.3389/fnbot.2022.1025817.
22. Zhong, J.; Long, T.; Sun, J.; Li, J.; Cao, Y. Delayed soft actor-critic based path planning method for UAV in dense obstacles environment. In Proceedings of the 2023 9th International Conference on Control Science and Systems Engineering (ICCSSE), Shenzhen, China, 16–18 June 2023; pp. 172–177. https://doi.org/10.1109/ICCSSE59359.2023.10245929.
23. Xie, Y.; Yu, C.; Zang, H.; Gao, F.; Tang, W.; Huang, J.; Chen, J.; Xu, B.; Wu, Y.; Wang, Y. Multi-UAV formation control with static and dynamic obstacle avoidance via reinforcement learning. In Proceedings of the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, 19–25 October 2025; pp. 20410–20417. https://doi.org/10.1109/IROS60139.2025.11246743.
24. He, H.; Yuan, W.; Chen, S.; Hou, Y.; Chen, S.; Tan, X.; Yang, J. Two-Timescale Trajectory Planning for UAV Formation Serving Hotspot in Unknown Environments With Complex Obstacles. *IEEE Trans. Cogn. Commun. Netw.* **2026**, *12*, 4261–4276. https://doi.org/10.1109/TCCN.2025.3637411.
25. Zhou, Y.; Kong, X.; Lin, K.P.; Liu, L. Novel Task Decomposed Multi-Agent Twin Delayed Deep Deterministic Policy Gradient Algorithm for Multi-UAV Autonomous Path Planning. *Knowl.-Based Syst.* **2024**, *287*, 111462. https://doi.org/10.1016/j.knosys.2024.111462.
26. Su, K.; Qian, F. Multi-UAV Cooperative Searching and Tracking for Moving Targets Based on Multi-Agent Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 11905. https://doi.org/10.3390/app132111905.
27. Liu, Y.; Liu, Z.; Wang, G.; Yan, C.; Wang, X.; Huang, Z. Flexible Multi-UAV Formation Control via Integrating Deep Reinforcement Learning and Affine Transformations. *Aerosp. Sci. Technol.* **2025**, *157*, 109812. https://doi.org/10.1016/j.ast.2024.109812.
28. 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.
29. Xing, X.; Zhou, Z.; Li, Y.; Xiao, B.; Xun, Y. Multi-UAV Adaptive Cooperative Formation Trajectory Planning Based on an Improved MATD3 Algorithm of Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12484–12499. https://doi.org/10.1109/TVT.2024.3389555.
30. Bui, D.N.; Phung, M.D.; Duy, H.P. Self-Reconfigurable V-Shape Formation of Multiple UAVs in Narrow Space Environments. In Proceedings of the 2024 IEEE/SICE International Symposium on System Integration (SII), Ha Long, Vietnam, 8–11 January 2024; pp. 1006–1011. https://doi.org/10.1109/SII58957.2024.10417519.
31. Fathian, K.; Summers, T.H.; Gans, N.R. Distributed formation control and navigation of fixed-wing UAVs at constant altitude. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 300–307. https://doi.org/10.1109/ICUAS.2018.8453462.

32. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114. https://doi.org/10.48550/arXiv.1312.6114.
33. Park, M.; Lee, S.Y.; Hong, J.S.; Kwon, N.K. Deep deterministic policy gradient-based autonomous driving for mobile robots in sparse reward environments. *Sensors* **2022**, *22*, 9574. https://doi.org/10.3390/s22249574.
34. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018*; Proceedings of Machine Learning Research; Dy, J., Krause, A., Eds.; PMLR: New York, NY, USA, 2018; Volume 80, pp. 1861–1870.
35. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.

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.