

Article

Dynamic Self-Organization and Safe Navigation for Hierarchical Embodied Swarms

Lanbo Wu  and Chen Wei 

National Key Laboratory of Aircraft Integrated Flight Control, School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing 100083, China; wulanbo@buaa.edu.cn

* Correspondence: weichen@buaa.edu.cn

Highlights

What are the main findings?

- A hierarchical embodied swarm framework with corridor-driven split–merge reconfiguration and feasibility projection enables coordinated multi-UAV navigation in complex environments.
- Multi-seed simulations show that LLM-assisted decisions remain feasible under the same projection layer and improve recovery in the most constrained alternating-gate scenario through stronger semantic split–merge reasoning.

What is the implication of the main finding?

- High-level semantic reasoning can be integrated into UAV swarms without sacrificing geometric and kinematic executability when explicit feasibility and safety constraints are enforced.
- The proposed framework provides a practical basis for multi-UAV search, inspection, and other missions in narrow passages and dense obstacle fields that require online reconfiguration and safe cooperative navigation.

Abstract

This paper is concerned with cooperative multi-UAV navigation in a planar obstacle environment. A hierarchical embodied swarm framework with leader, subleader, and follower roles is proposed. At the high level, a passable-corridor-driven decision layer is developed to perform split–merge reconfiguration and navigate/encircle mode switching. At the low level, a multi-term force synthesis controller is constructed for formation maintenance, inter-agent collision avoidance, obstacle avoidance, and sub-swarm cohesion. To accommodate both rule-based and local large language model (LLM) decisions, a feasibility projection operator is introduced so that only kinematically admissible structural actions are executed. In addition, a LiDAR-based obstacle-repulsion term and an occlusion-attenuated attraction mechanism are incorporated to improve navigation safety in cluttered environments. A Lyapunov analysis of the smooth controller core further certifies that, for a known (possibly time-varying) cruise velocity compensated by feedforward, the formation tracking error is uniformly bounded by the initial energy. Finally, multi-seed numerical simulations verify the proposed framework in standard, ablated, and complex scenarios. In the hardest alternating-gate scenario, the LLM-assisted variant raises mission success from 0.000 to 0.100, increases the goal-reaching ratio from 0.025 to 0.125, and reduces the mean terminal error from 44.738 m to 39.851 m, showing the value of semantic high-level reconfiguration under tight passage constraints.



Academic Editor: Xingling Shao

Received: 19 May 2026

Revised: 5 June 2026

Accepted: 8 June 2026

Published: 10 June 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: embodied swarm; hierarchical control; dynamic split–merge; self-organization; LLM-constrained decision

1. Introduction

Unmanned aerial vehicle (UAV) swarms have become a key enabler for missions that demand wide-area coverage, fault tolerance, and parallel task execution, including search and rescue, reconnaissance, environmental monitoring, and infrastructure inspection [1–3]. A single platform cannot match the spatial reach and redundancy that a coordinated group provides, which has driven sustained research interest in multi-UAV cooperative systems.

In recent years, the field has progressed from early distributed flocking models and self-driven particle systems [4,5] toward mission-capable swarm autonomy in complex environments. However, real-world deployments introduce a qualitatively harder class of problems: when a swarm operates in narrow passages, cluttered urban scenes, or dynamically changing task areas, it may need to complete multiple sub-tasks to achieve the final goal. Satisfying these multi-stage demands goes well beyond what single-stage reactive swarm designs can offer: the swarm topology must be reconfigured online in response to corridor geometry, individual-level safety must be maintained throughout structural transitions, and mission-mode switching must be coordinated through a decision hierarchy that purely reactive distributed architectures do not provide. Three interconnected challenges therefore arise: (i) how to organize a role hierarchy that drives corridor-aware split–merge reconfiguration; (ii) how to guarantee safe passage through narrow and cluttered environments during structural transitions; and (iii) how to incorporate semantic reasoning without sacrificing geometric and kinematic executability.

Swarm self-organization and structural reconfiguration. Existing studies have investigated social-learning-based organization [6], finite-time observers [7], informed-agent mechanisms for preventing splitting [8], experience replay [9], split–merge reconfiguration [10], heuristic strategy [11], bio-inspired fission–fusion control [12], event-triggered self-organizing control [13], transferable communication modules [14], improved potential field function [15], “nervous systems” such as self organizing nervous systems [16] and topology-driven trajectory optimization [17]. These results demonstrate diverse coordination capabilities, yet each method addresses only a subset of the challenges involved in complex multi-stage swarm navigation. Split–merge reconfiguration [10] handles formation-level restructuring in dynamic environments but relies on pre-specified waypoints and does not use corridor passability as the trigger for structural decisions, nor does it address mission-mode switching at the task level. Bio-inspired fission–fusion control [12] achieves dynamic group reassignment through local interaction rules but provides no mechanism to verify the geometric or kinematic feasibility of structural transitions before executing them in narrow passages. Self-organizing nervous systems [16] assign specialized roles for distributed aggregation and actuation, but their design targets open-environment deployments without accounting for UAV kinematic constraints or dense-obstacle safety during reconfiguration. Topology-driven trajectory optimization [17] finds collision-free paths by exploring topological variants but operates on individual trajectories and does not extend to coordinated multi-agent structural decisions. More broadly, methods addressing formation cohesion [8], event-triggered coordination [13], and heuristic coverage [11] each focus on isolated aspects of swarm coordination in single-stage or simplified task settings. Recent robust event-triggered path-following studies with output constraints also indicate that update-on-demand control

and constrained-error transformations can reduce command update frequency while keeping tracking errors within prescribed bounds under disturbances [18]. These ideas are highly relevant to UAV swarms, but most existing formulations still focus on single-vehicle path following or fixed formation tracking rather than online split–merge decisions coupled with local obstacle constraints. Consequently, an architectural logic that unifies corridor-driven split–merge decisions, role-hierarchy maintenance, safe passage negotiation, and multi-stage mission-mode switching within one closed-loop decision framework has not been established.

Embodied intelligence for aerial swarms. When embodied intelligence is introduced into UAV systems, the focus shifts to closing the perception–decision–action loop in the physical world. Recent studies have reviewed learning-based motion planning and control from the perspective of embodied intelligence [19] and discussed the transition from single-agent embodiment to multi-agent embodied AI [20]. In aerial scenarios, representative works include vision-and-language navigation [21], embodied UAV benchmarks [22], real-world language-conditioned UAV imitation benchmarks [23], and decentralized reinforcement learning for scalable embodied robotic swarms [24]. These works have pushed aerial embodiment from passive perception toward interactive decision making, and some of them have started to consider multi-agent scalability. Nevertheless, critical limitations persist across these directions. Language-and-vision navigation approaches such as AerialVLN [21] and benchmark platforms [22,23] address the perception–language–action loop for individual aerial agents but are inherently single-platform approaches, leaving multi-agent role assignment, topology preservation, and cooperative corridor negotiation outside their scope. Decentralized reinforcement learning for embodied swarms [24] scales cooperative behavior across multiple agents through learned policies, yet such policies provide no hard guarantee on geometric safety or structural feasibility when the swarm must reorganize online in narrow constrained passages. The transition from single-agent embodied perception to multi-agent embodied swarm control with explicit role hierarchies and safety-constrained structural evolution therefore remains an open challenge.

LLM-based semantic decision making for swarms. Recently, LLM-based decision making has opened a new route for high-level autonomy. Language-grounded decision systems such as SayCan and Inner Monologue [25,26] show that semantic reasoning can improve task decomposition and planning, and SMART-LLM, MUTP-LLM, and unified task–spatial UAV decision systems have further explored multi-robot or swarm-oriented settings [27–29]. In parallel, swarm decision-making studies have investigated distributed situation awareness [30], information-fusion decision making [31], communication information aggregation [32], confrontation strategies [33–36], and pigeon-inspired optimization [37]. The above results are encouraging, but a critical gap remains. SayCan [25] and Inner Monologue [26] ground language in robot affordances and closed-loop feedback for single-platform manipulation or navigation but do not manage multi-agent structural states. SMART-LLM [27] extends language-based task decomposition to multi-robot teams but assumes stable formation configurations, sidestepping the dynamic structural transitions required for swarm corridor navigation. MUTP-LLM [28] applies LLM reasoning to multi-UAV task allocation but focuses on mission-level assignment rather than structural transitions; the geometric and kinematic feasibility of split-count selection and sub-swarm sizing relative to corridor width is left unverified. Classical swarm decision methods [30–32] handle distributed information processing and situational reasoning effectively but rely on handcrafted logic or scenario-specific learned policies that do not naturally accommodate semantic flexibility. In either case, the connection between top-layer semantic decisions and bottom-layer executable swarm control is still weak, and hard geometric, kinematic, and safety constraints are rarely enforced through an explicit feasibility

projection. This issue becomes particularly significant when complex maneuvers must be generated online in cluttered environments.

Although the above studies have laid important foundations, they remain fragmented across self-organization, embodiment, and semantic decision making. In response to the above discussions, this paper studies dynamic self-organization and safe navigation for hierarchical embodied swarms in obstacle-rich environments. The first motivation is to establish a clear swarm decision hierarchy that can organize leader–subleader–follower cooperation together with split–merge reconfiguration and task-mode switching. The second motivation is to develop a safe navigation mechanism that remains effective when the swarm passes through narrow and branching passages, encounters concave obstacles, and must preserve control executability during structural evolution. The third motivation is to exploit the semantic flexibility of LLMs without giving up rule-level safety and feasibility. To this end, we develop a closed-loop framework that connects corridor perception, hierarchical decision, rule-constrained LLM assistance, and low-level safe control within one embodied swarm system.

The main contributions of the paper are as follows:

- **Hierarchical swarm decision architecture and rule-based mode design:** A hierarchical embodied swarm architecture with leader, subleader, and follower roles is established, together with a rule-based mode design that organizes corridor negotiation, split–merge reconfiguration, and task-mode switching in a unified decision loop.
- **Safe navigation mechanism for cluttered environments:** A safety-oriented navigation mechanism is developed, including LiDAR-based obstacle repulsion, occlusion-aware attraction regulation, and geometric safety projection, so that the swarm can maintain coordinated motion in narrow and obstacle-dense environments.
- **LLM-assisted decision making under rule constraints:** An LLM-assisted top-layer decision interface is introduced to connect semantic reasoning with swarm structural control. By combining language-guided decision proposals with rule constraints and feasibility projection, only geometrically and kinematically executable actions are delivered to the bottom-layer controller. The multi-seed complex-scenario evaluation further shows that this semantic layer is most useful when repeated split–merge reasoning is required, as in the narrow alternating gate benchmark.

2. Notation and System Modeling

2.1. Notation

To avoid symbol conflicts, the manuscript adopts the following conventions:

- The total number of UAVs is N , and the UAV indices are $i, j \in \{1, \dots, N\}$.
- Continuous time is denoted by $t \geq 0$; the high-level decision layer is updated at discrete epochs indexed by $k \in \mathbb{N}^+$ with sampling step Δt .
- The number of sub-swarms is $S(k)$, and the partition set is $\mathbb{S}(k)$.
- The s th sub-swarm is denoted by $\mathcal{S}_s(k)$ with $s \in \{1, \dots, S(k)\}$.
- Vectors are boldface: position, velocity, acceleration, and control input are denoted as $\mathbf{p}$, $\mathbf{v}$, $\mathbf{a}$, and $\mathbf{u}$, respectively.

The core symbols used throughout the paper are summarized in Table 1.

Table 1. Core symbols.

Symbol	Meaning
N	Total number of UAVs.
$t; k, \Delta t$	Continuous time; discrete decision epoch and sampling step.
$\mathbf{p}_i(t), \mathbf{v}_i(t), \dot{\mathbf{v}}_i(t)$	Position, velocity, and acceleration of UAV i .
$\mathbf{u}_i, m_i$	Control input and mass of UAV i .
$S(k), \mathbb{S}(k), \mathcal{S}_s(k)$	Number of sub-swarms, partition set, and sth sub-swarm.
$\ell_s(k), \mathbf{c}_s(k)$	Subleader index and local waypoint of sub-swarm s .
$\mathbf{p}_g$	Global target point.
$\bar{\mathbf{p}}_s(k), \bar{\mathbf{p}}(k)$	Centroid of sub-swarm s and global centroid.

2.2. Hierarchical Organization and State Reporting

The sub-swarm partition at time k is defined as

$$\mathbb{S}(k) = \{\mathcal{S}_1(k), \dots, \mathcal{S}_{S(k)}(k)\}, \quad \bigcup_{s=1}^{S(k)} \mathcal{S}_s(k) = \{1, \dots, N\}, \quad \mathcal{S}_a(k) \cap \mathcal{S}_b(k) = \emptyset \quad (a \neq b), \quad (1)$$

where $\mathcal{S}_s(k)$ is the member set of sub-swarm s and $S(k)$ is the total number of sub-swarms. The centroid of sub-swarm s and the global centroid are

$$\bar{\mathbf{p}}_s(k) = \frac{1}{|\mathcal{S}_s(k)|} \sum_{i \in \mathcal{S}_s(k)} \mathbf{p}_i(k), \quad \bar{\mathbf{p}}(k) = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i(k), \quad (2)$$

where $|\mathcal{S}_s(k)|$ is the size of sub-swarm s .

The subleader report packet (Pack) is defined by

$$\mathcal{P}_s(k) = \{(j, \mathbf{p}_j(k), \mathbf{v}_j(k)) \mid j \in \mathcal{S}_s(k)\}, \quad (3)$$

where $\mathcal{P}_s(k)$ is the reported state set of sub-swarm s at time k .

2.3. UAV Dynamics

A single UAV follows a damped continuous-time double-integrator model:

$$\dot{\mathbf{p}}_i(t) = \mathbf{v}_i(t), \quad (4)$$

$$\dot{\mathbf{v}}_i(t) = \frac{\mathbf{u}_i(t)}{m_i} - c_d \mathbf{v}_i(t), \quad (5)$$

where $c_d = 0.5$ is the linear damping coefficient, $\mathbf{u}_i$ is the control input, and m_i is the mass. By choosing the control input as

$$\mathbf{u}_i = m_i (\mathbf{a}_i^{\text{cmd}} + c_d \mathbf{v}_i) \quad (6)$$

where $\mathbf{a}_i^{\text{cmd}}$ is the desired acceleration, Equation (5) can be transformed to $\dot{\mathbf{v}}_i = \mathbf{a}_i^{\text{cmd}}$. For execution, the model is integrated at a fixed step Δt .

2.4. LiDAR Perception Model

Each UAV is equipped with a planar LiDAR sensor mounted at the fixed flight height $z = z_{\text{fix}}$. Let $\Delta\theta$ be the angular resolution and R_L the sensing range. The scan-ray set is

$$\Theta = \{\theta_r = (r-1)\Delta\theta \mid r = 1, \dots, n_\theta\}, \quad n_\theta = \left\lfloor \frac{2\pi}{\Delta\theta} \right\rfloor, \quad (7)$$

which covers $[0, 2\pi)$. For control and corridor extraction, the controller uses the fused distance ring

$$d_{i,r}(k) = \min_{o \in \mathcal{O}} \tilde{d}_{i,r}^{(o)}(k), \quad r = 1, \dots, n_\theta, \quad (8)$$

where $\tilde{d}_{i,r}^{(o)}(k) \in [0, R_L]$ is the range returned by obstacle o along ray r . The same fused ring is used by the high-level corridor detector and the low-level obstacle-avoidance controller.

3. High-Level Dynamic Self-Organization and Decision Making

3.1. Action Sets and Task Modes

The structural and task-level action sets are

$$\mathcal{A}_{\text{str}} = \{\text{split}, \text{merge}, \text{keep}\}, \quad \mathcal{A}_{\text{task}} = \{\text{navigate}, \text{encircle}\}, \quad (9)$$

where $\mathcal{A}_{\text{str}}$ is for sub-swarm reconfiguration and $\mathcal{A}_{\text{task}}$ is for behavioral mode selection. For sub-swarm $\mathcal{S}_s(k)$, define target proximity as

$$\delta_s(k) = \min(d_s^c(k), d_s^\ell(k)), \quad (10)$$

where

$$d_s^c(k) = \|\bar{\mathbf{p}}_s(k) - \mathbf{p}_g\|_2, \quad d_s^\ell(k) = \|\mathbf{p}_{\ell_s(k)}(k) - \mathbf{p}_g\|_2. \quad (11)$$

Here, $d_s^c(k)$ is the centroid-to-goal distance and $d_s^\ell(k)$ is the subleader-to-goal distance. A hysteresis switching rule is used:

$$\sigma_s(k) = \begin{cases} \text{encircle}, & \delta_s(k) < r_{\text{in}}, \\ \text{navigate}, & \delta_s(k) \geq r_{\text{out}}, \quad r_{\text{out}} > r_{\text{in}}, \\ \sigma_s(k-1), & \text{otherwise,} \end{cases} \quad (12)$$

where $\sigma_s(k)$ is the task mode, while r_{in} and r_{out} are the entry and exit thresholds.

3.2. Rule-Based Decisions: Split and Merge

Each subleader uses the LiDAR distance ring defined above to extract candidate passable corridors leading to the target. The detailed gap-detection logic follows the local-minimum pairing strategy in [38]. After path detection, the swarm decides whether to split into multiple sub-swarms for exploration or merge together. The split feasibility condition is

$$S(k) = 1, \quad n_c(k) \geq 2, \quad k - k_{\text{last}} \geq T_{\text{cd}}, \quad \|\bar{\mathbf{p}}(k) - \mathbf{p}_g\|_2 \geq r_{\text{merge}}, \quad (13)$$

where k_{last} is the latest structural-change time, T_{cd} is the cooldown duration, r_{merge} is the near-goal merge radius, and $n_c(k)$ is the number of feasible paths. The minimum inter-sub-swarm distance is

$$d_{\text{swarm}}^{\min}(k) = \min_{a \neq b} \|\bar{\mathbf{p}}_a(k) - \bar{\mathbf{p}}_b(k)\|_2, \quad (14)$$

where $a, b \in \{1, \dots, S(k)\}$. The number of near-goal sub-swarms is

$$n_{\text{near}}(k) = |\{s \mid \|\bar{\mathbf{p}}_s(k) - \mathbf{p}_g\|_2 < r_{\text{merge}}\}|, \quad (15)$$

where $n_{\text{near}}(k)$ counts sub-swarms in the goal neighborhood. Merging is feasible if

$$n_{\text{near}}(k) \geq 2 \quad \text{or} \quad d_{\text{swarm}}^{\min}(k) < d_{\text{merge}}, \quad (16)$$

where d_{merge} is the merge-distance threshold.

3.3. LLM Decision and Feasibility Projection

Define rule-feasibility indicators as

$$\mathbb{I}_{\text{split}}(k) = \begin{cases} 1, & \text{split feasible,} \\ 0, & \text{otherwise,} \end{cases} \quad \mathbb{I}_{\text{merge}}(k) = \begin{cases} 1, & \text{merge feasible,} \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where $\mathbb{I}_{\text{split}}(k), \mathbb{I}_{\text{merge}}(k) \in \{0, 1\}$. The LLM input state vector is

$$\mathbf{z}(k) = \begin{bmatrix} k, \Delta t & S(k), n_c(k) & w_{\max}(k), w_{\min}(k) & \bar{d}_g(k) \\ n_{\text{near}}(k) & d_{\text{swarm}}^{\min}(k) & \mathbb{I}_{\text{split}}(k) & \mathbb{I}_{\text{merge}}(k) \end{bmatrix}^{\top}, \quad (18)$$

where

$$\bar{d}_g(k) = \frac{1}{S(k)} \sum_{s=1}^{S(k)} \|\bar{\mathbf{p}}_s(k) - \mathbf{p}_g\|_2, \quad (19)$$

and $w_{\max}(k), w_{\min}(k)$ are the maximum and minimum candidate corridor widths.

Let the LLM inference model be F_{θ} and the prompt constructor be $P(\cdot)$. Then,

$$(\hat{a}_k, \hat{r}_k) = F_{\theta}(P(\mathbf{z}(k))), \quad \hat{a}_k \in \mathcal{A}_{\text{str}}, \quad (20)$$

where $\hat{a}_k$ is the suggested structural action and $\hat{r}_k$ is the textual rationale. The LLM trigger indicator is

$$\mathbb{I}_{\text{llm}}(k) = \begin{cases} 1, & k = 1 \text{ or } k - k_{\text{llm}} \geq T_{\text{llm}}, \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

where k_{llm} is the previous LLM invocation step and T_{llm} is the invocation interval. The candidate action is

$$a_k = \begin{cases} \hat{a}_k, & \mathbb{I}_{\text{llm}}(k) = 1, \\ a_k^{\text{rule}}, & \mathbb{I}_{\text{llm}}(k) = 0, \end{cases} \quad (22)$$

where $a_k^{\text{rule}} \in \mathcal{A}_{\text{str}}$ is the rule-based action. When the LLM is not available or we decide to only adopt the rule-based decision, let $a_k = a_k^{\text{rule}}$. The executed action is obtained by feasibility projection:

$$a_k^* = \Pi_{\text{feas}}(a_k, \mathbf{z}(k)) = \begin{cases} \text{split}, & a_k = \text{split} \wedge \mathbb{I}_{\text{split}}(k) = 1, \\ \text{merge}, & a_k = \text{merge} \wedge \mathbb{I}_{\text{merge}}(k) = 1, \\ \text{keep}, & \text{otherwise,} \end{cases} \quad (23)$$

where a_k^* is the feasible executable structural action. If the LLM back end fails (timeout, empty response, or parsing failure), the system falls back to a_k^{rule} .

The LLM module runs on a locally hosted Ollama server, requiring no external network access. Two open-source quantized models are evaluated: qwen2.5:3b (≈ 3 B parameters) and qwen3.5:4b (≈ 4 B parameters); models with no more than ≈ 10 B parameters are preferred to keep median inference latency within one navigation sub-horizon on a single consumer GPU. The decoding temperature is fixed at $T = 0$ (greedy decoding) to produce deterministic structural outputs; the maximum output length is 64 tokens, which is sufficient for the constrained JSON response. The LLM is queried every $T_{\text{llm}} = 50$ steps (5 s at $\Delta t = 0.1$ s), while the low-level force controller runs at the full 10 Hz rate; a per-call timeout of 120 s is enforced.

The prompt $P(z(k))$ has a fixed three-part structure: (i) a role preamble identifying the model as a “swarm navigation high-level decision planner”; (ii) three explicit decision rules linking the state fields {can_split, path_count, avg_goal_distance, can_merge, near_goal_subswarms} to preferred actions; and (iii) a strict output constraint requiring a single JSON object {“decision”:..., “reason”:...}. The parser first extracts a JSON block via regex; on failure, it falls back to a keyword search for split/merge/keep; the default on any parsing failure is the conservative keep.

4. Low-Level Controller Design

4.1. Reference Trajectory and Attraction Term

In navigation mode, for $i \in \mathcal{S}_s(k)$ with formation index r_i , the reference position and velocity are

$$\mathbf{p}_i^*(k) = \mathbf{c}_s(k) + \alpha_s(k) \Delta \mathbf{f}_{r_i}, \quad \mathbf{v}_i^*(k) = v_s(k) \frac{\mathbf{p}_i^*(k) - \mathbf{p}_i(k)}{\|\mathbf{p}_i^*(k) - \mathbf{p}_i(k)\|_2 + \epsilon_v}, \quad (24)$$

where $\Delta \mathbf{f}_{r_i}$ is the formation-template offset, $\alpha_s(k)$ is the formation scale, $v_s(k)$ is the desired cruising speed of sub-swarm s , and $\epsilon_v > 0$ avoids division by zero.

The attraction term is

$$\mathbf{f}_{\text{att},i} = k_p(\mathbf{p}_i^* - \mathbf{p}_i) + k_v(\mathbf{v}_i^* - \mathbf{v}_i), \quad (25)$$

where $k_p, k_v > 0$ are the position and velocity error gains.

4.2. Separation, Cohesion, and Following Coupling

The global separation term is

$$\mathbf{f}_{\text{sep},i} = \sum_{j \neq i, d_{ij} < r_{\text{sep}}} k_{\text{sep}} \frac{\mathbf{p}_i - \mathbf{p}_j}{d_{ij}^2}, \quad d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2, \quad (26)$$

where r_{sep} is the separation activation radius and $k_{\text{sep}} > 0$ is the separation gain.

The nearest-neighbor intra-sub-swarm attraction is

$$\mathbf{f}_{\text{intra},i} = \begin{cases} k_{\text{intra}}(d_{\text{nn},i} - d_{\text{thr}})\hat{\mathbf{e}}_{\text{nn},i}, & d_{\text{nn},i} > d_{\text{thr}}, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (27)$$

where $d_{\text{nn},i}$ is the nearest-neighbor distance of UAV i within its sub-swarm, $\hat{\mathbf{e}}_{\text{nn},i}$ is the corresponding unit direction, and d_{thr} is the activation threshold.

The follower–subleader coupling term is

$$\mathbf{f}_{\text{fol},i} = k_{\ell} \max(0, d_{i\ell} - d_0)\hat{\mathbf{e}}_{i\ell} + k_{\ell v}(\mathbf{v}_{\ell_s(k)} - \mathbf{v}_i), \quad (28)$$

where $d_{i\ell} = \|\mathbf{p}_i - \mathbf{p}_{\ell_s(k)}\|_2$ is the follower–subleader distance, $\hat{\mathbf{e}}_{i\ell}$ is the corresponding unit direction, and d_0 is the following activation threshold.

The centroid cohesion term is

$$\mathbf{f}_{\text{coh},i} = k_{\text{coh}} \max(0, d_{ic} - d_c)\hat{\mathbf{e}}_{ic}, \quad d_{ic} = \|\mathbf{p}_i - \bar{\mathbf{p}}_s\|_2, \quad (29)$$

where d_c is the cohesion activation distance and $\hat{\mathbf{e}}_{ic}$ is the unit vector pointing to the sub-swarm centroid.

When the segment between UAV i and its attraction target—formation slot p_i^* , nearest intra-swarm neighbor $p_{nn,i}$, subleader p_{ℓ_s} , or sub-swarm centroid $\bar{p}_s$ —crosses an obstacle, unreduced attraction pulls the UAV toward the wall. A unified occlusion attenuation factor is therefore applied to all inter-agent and agent-to-target attraction terms:

$$\sigma(p_i, q) = \begin{cases} \sigma_{\text{occ}}, & \text{segment } \overline{p_i q} \text{ intersects any obstacle,} \\ 1, & \text{otherwise,} \end{cases} \quad (30)$$

where $\sigma_{\text{occ}} \in (0, 1)$ is the occlusion scale (set to 0.2 in practice). Concretely, the horizontal components of $f_{\text{att},i}$ are multiplied by $\sigma(p_i, p_i^*)$; $f_{\text{intra},i}$ by $\sigma(p_i, p_{nn,i})$; $f_{\text{fol},i}$ by $\sigma(p_i, p_{\ell_s})$; and $f_{\text{coh},i}$ by $\sigma(p_i, \bar{p}_s)$. As shown in Figure 1, this mechanism substantially reduces wall-collision incidents caused by occluded attraction forces in complex multi-obstacle environments.

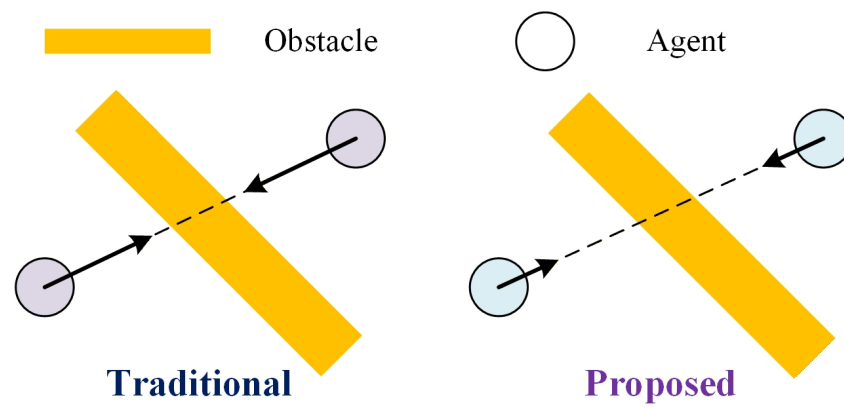


Figure 1. Line-of-sight occlusion attenuation. The orange bar denotes an obstacle and the circles denote agents; the solid arrow marks the attraction force that is actually applied, while the dashed segment marks the portion of the line of sight crossing the obstacle. In the traditional scheme (left), the attraction acts at full strength even when the connecting segment is occluded, whereas in the proposed scheme (right), the occluded attraction component is attenuated according to (30).

The intra-sub-swarm short-range repulsion prevents excessive crowding between members of the same sub-swarm:

$$f_{\text{intra,rep},i} = \sum_{j \in \mathcal{S}_s(k), d_{ij} < d_{\text{ir}}} k_{\text{ir}} \frac{p_i - p_j}{d_{ij}^2}, \quad (31)$$

where d_{ir} is the intra-sub-swarm repulsion radius and $k_{\text{ir}} > 0$ is its gain. Unlike the global separation term $f_{\text{sep},i}$ (which acts on all neighbors), this term is restricted to same-sub-swarm members with a shorter activation range, maintaining uniform intra-formation spacing.

A corridor centering term biases each UAV toward the sub-swarm waypoint $c_s(k)$ during navigation, keeping the formation aligned with the center of the passable corridor rather than drifting sideways:

$$f_{\text{path},i} = k_{\text{path}}(c_s(k) - p_{i,xy}), \quad (32)$$

where $k_{\text{path}} > 0$ is the path-center gain.

4.3. LiDAR-Based Obstacle Repulsion

The obstacle force is computed from the fused LiDAR distance ring. Define the active local-minimum set

$$\mathcal{L}_i(k) = \{r \mid d_{i,r}(k) < d_{i,r\ominus 1}(k) \wedge d_{i,r}(k) < d_{i,r\oplus 1}(k) \wedge d_{i,r}(k) < R_t\}, \quad (33)$$

where R_t is the avoidance trigger radius and $\ominus, \oplus$ denote circular indexing on the ring. For each $r \in \mathcal{L}_i(k)$, the outward direction is

$$\hat{e}_{\text{away},i,r}(k) = \begin{bmatrix} -\cos \theta_r \\ -\sin \theta_r \end{bmatrix}. \quad (34)$$

Define the repulsive magnitude associated with ray r as

$$\omega_{i,r}(k) = k_{\text{rep}} \frac{\max(0, 1/d_{i,r}(k) - 1/R_t)}{d_{i,r}(k)^2} \left(1 + k_b \frac{\max(0, R_t - d_{i,r}(k))}{R_t}\right)^\gamma \quad (35)$$

$$+ \mathbb{I}_{\{d_{i,r}(k) < d_{\text{safe}}\}} k_e (d_{\text{safe}} - d_{i,r}(k) + 1)^2 \quad (36)$$

and the obstacle repulsion force by

$$f_{\text{obs},i}(k) = \sum_{r \in \mathcal{L}_i(k)} \omega_{i,r}(k) \hat{e}_{\text{away},i,r}(k), \quad (37)$$

where $k_{\text{rep}} > 0$ is the repulsion gain, $k_b > 0$ and $\gamma \geq 1$ adjust the short-range enhancement, d_{safe} is the safety distance, $k_e > 0$ is the near-obstacle compensation gain, and $\mathbb{I}_{\{\cdot\}}$ is the indicator function. If $\mathcal{L}_i(k) = \emptyset$, then $f_{\text{obs},i}(k) = \mathbf{0}$.

4.4. Safe Navigation Pipeline

The total desired acceleration is

$$a_i^{\text{cmd}} = f_{\text{att},i} + f_{\text{sep},i} + f_{\text{path},i} + f_{\text{intra},i} + f_{\text{intra,rep},i} + f_{\text{fol},i} + f_{\text{coh},i} + f_{\text{obs},i}, \quad (38)$$

where the eight terms correspond to attraction, global separation, corridor centering, intra-swarm cohesion, intra-swarm short-range repulsion, following coupling, centroid cohesion, and obstacle avoidance. The terms $f_{\text{path},i}$ and $f_{\text{intra,rep},i}$ are active only in navigate mode and set to $\mathbf{0}$ in encircle mode.

Before execution, saturation is applied:

$$\|a_{i,xy}^{\text{cmd}}\|_2 \leq a_{xy}^{\text{max}}, \quad (39)$$

where a_{xy}^{max} represents the acceleration limits. Constraint handling is therefore layered: the high-level feasibility projection rejects infeasible split-merge actions, the LiDAR repulsion and occlusion attenuation terms regulate obstacle and line-of-sight risk, and acceleration saturation keeps commands actuator-feasible. Decision-layer logic is event-triggered (LLM interval, split-merge and follower-reassignment cooldowns), while the low-level safety controller runs at every integration step. Classical Lyapunov analyses establish boundedness, cohesion, and collision avoidance for the smooth, continuous-time, purely potential-based sub-class of swarm controllers, e.g., attraction–repulsion aggregation [39] and potential-field flocking with switching-topology guarantees [40,41]. In the same spirit, we isolate a fixed-structure nominal core—the conservative part of the force synthesis plus local velocity damping—and analyze it with a mechanical-energy Lyapunov function, which yields the following boundedness certificate.

Theorem 1 (Uniform error boundedness under a known time-varying cruise). *Consider the smooth conservative core of the synthesized controller (38)—its seven differentiable force terms, with the discontinuous LiDAR repulsion excluded—on a structure-frozen window $\mathcal{W} = [t_0, t_1)$, and suppose the formation cruises rigidly at a common, possibly time-varying velocity $\mathbf{v}_s(t)$ whose acceleration $\dot{\mathbf{v}}_s$ is known and applied as feedforward. Let $\mathbf{e}_i := \mathbf{p}_i - \mathbf{p}_i^*(t)$ be the formation tracking error and V the closed-loop mechanical energy (kinetic energy of the error plus the aggregate potential). Then V is non-increasing, and the formation error is uniformly bounded by its initial value,*

$$\|\mathbf{e}_i(t)\| \leq \sqrt{\frac{2V(t_0)}{k_p}}, \quad \forall t \in \mathcal{W}, \forall i. \quad (40)$$

Moreover, if the frozen smooth regime persists, LaSalle's invariance principle yields asymptotic convergence: the error velocity satisfies $\dot{\mathbf{e}}_i \rightarrow \mathbf{0}$, and whenever the desired formation is the isolated critical point of the aggregate potential reached by the trajectory, the tracking error itself vanishes, $\mathbf{e}_i(t) \rightarrow \mathbf{0}$ for all i .

The precise smooth-core reduction and the energy argument are given in Appendix B. At each discrete step, the framework executes the following items:

1. Collect Pack data from each sub-swarm and update high-level state summaries.
2. Generate structural candidate actions via rules or the LLM, then project to feasible action $\mathbf{a}_k^*$.
3. Update task mode (navigate/encircle), sub-swarm waypoints, and reference formations; execute dynamic follower reassignment (see below).
4. Synthesize low-level control terms and apply acceleration saturation.

After a split, some followers may be geometrically far from their assigned subleader, causing formation dispersion. At each planner update, a dynamic follower reassignment step evaluates whether any follower should be transferred to a different sub-swarm. For follower i (subleaders are protected from reassignment), the assignment cost in sub-swarm s is

$$\phi_s(i) = w_\ell d_{i\ell_s} + w_c d_{ic_s}, \quad (41)$$

where $d_{i\ell_s} = \|\mathbf{p}_i - \mathbf{p}_{\ell_s}\|_2$ is the distance to the subleader and $d_{ic_s} = \|\mathbf{p}_i - \mathbf{c}_s\|_2$ is the distance to the waypoint, with weights $w_\ell, w_c > 0$. Follower i is moved to sub-swarm s' if the improvement $\Delta\phi = \phi_s(i) - \phi_{s'}(i) \geq \Delta_{\min}$. Each update is limited to at most $N_{\text{move}}^{\max}$ transfers, and a per-follower cooldown of τ_{ra} steps prevents rapid oscillation. The overall structure of the proposed framework is shown in Figure 2a. The algorithm flow chart is shown in Figure 2b.

From an implementation perspective, the hierarchy can be deployed through leader–subleader communication rather than all-to-all exchange: subleaders transmit compact Pack-level summaries, while followers run the low-level controller using onboard localization, velocity estimates, and LiDAR scans. The LLM module is outside the hard real-time loop; any timeout, invalid output, or infeasible suggestion is replaced by the rule-based action through feasibility projection, and the saturated acceleration command can be converted to velocity or attitude references for the UAV autopilot.

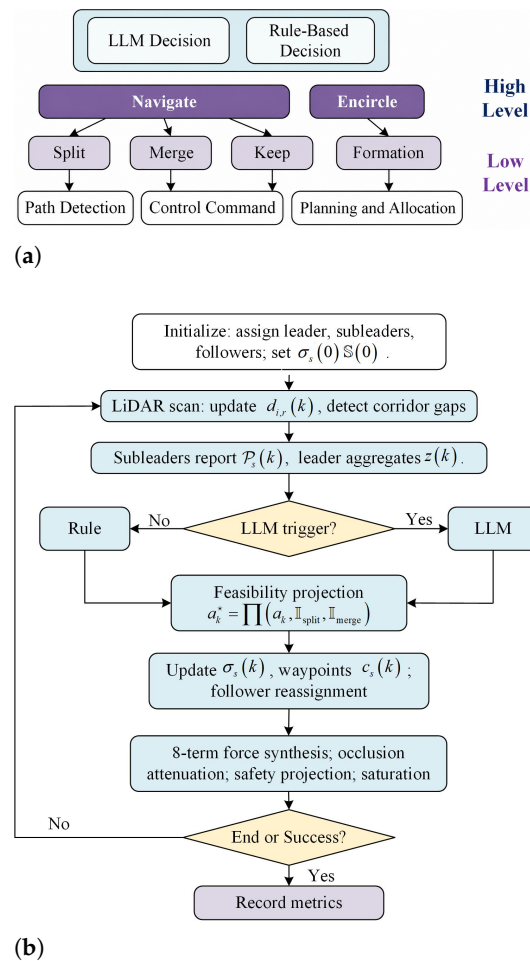


Figure 2. Framework architecture and execution flow. (a) Overall structure of the proposed hierarchical embodied swarm framework, showing the coupling among corridor perception, the leader-subleader-follower decision hierarchy, rule-constrained LLM assistance, and the low-level safe controller; (b) algorithm flow chart of the same framework, detailing the per-step execution order from perception and high-level decision through feasibility projection to low-level control.

5. Simulation and Analysis

5.1. Protocol and Metric Definitions

All simulations use the same simulator and are evaluated over multiple random initializations. Simulation 1 uses ten random seeds $\{1, \dots, 10\}$ for each decision variant, Simulation 2 uses ten seeds $\{1, \dots, 10\}$ for the controller ablation, and Simulation 3 uses ten seeds $\{1, \dots, 10\}$ for each rule/LLM scenario pair. The common setup is $N = 20$, $\Delta t = 0.1$ s. Simulations use a total horizon $T = 100$ s ($K = 1000$ steps). For every variant or scenario v , let $\mathcal{R}_v$ denote its run set with cardinality $n_v = |\mathcal{R}_v|$, where $n_v = 10$ in Simulation 1, $n_v = 10$ in Simulation 2, and $n_v = 10$ in Simulation 3.

For run $r \in \mathcal{R}_v$, define the terminal goal error of UAV i as

$$e_i^{(r)} = \left\| p_i^{(r)}(K) - p_g^{(r)} \right\|_2. \quad (42)$$

Here, $p_i^{(r)}(K)$ is the final position of UAV i at step K in run r , and $p_g^{(r)}$ is the goal position used in run r .

The mean and tail terminal errors are

$$E_{\text{mean}}^{(r)} = \frac{1}{N} \sum_{i=1}^N e_i^{(r)}, \quad E_{90}^{(r)} = \text{Perc}_{90} \left(\{e_i^{(r)}\}_{i=1}^N \right), \quad (43)$$

where $\text{Perc}_{90}(\cdot)$ denotes the 90th percentile operator.

Let r_{succ} be the single-UAV success radius ($r_{\text{succ}} = 8$ m). The goal-reaching ratio and mission success indicator are

$$R_{\text{goal}}^{(r)} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(e_i^{(r)} \leq r_{\text{succ}}), \quad Y^{(r)} = \mathbf{1}(R_{\text{goal}}^{(r)} \geq \tau_{\text{ms}}), \quad (44)$$

where $\mathbf{1}(\cdot)$ is the indicator function and $\tau_{\text{ms}} = 0.8$ is the mission-level threshold.

Define the minimum clearance around obstacles at step k as

$$d_{\min}^{(r)}(k) = \min_{i \in \{1, \dots, N\}} d_i^{\text{obs},(r)}(k), \quad (45)$$

where $d_i^{\text{obs},(r)}(k)$ is the distance from UAV i to the nearest obstacle in run r at step k . Let d_{safe} denote the configured safety distance. The clearance-pressure ratio and minimum clearance are

$$R_{\text{unsafe}}^{(r)} = \frac{1}{K-1} \sum_{k=1}^{K-1} \mathbf{1}(d_{\min}^{(r)}(k) < d_{\text{safe}}), \quad d_{\text{clr,min}}^{(r)} = \min_{1 \leq k \leq K-1} d_{\min}^{(r)}(k). \quad (46)$$

For trajectory efficiency, the per-UAV path length and its swarm average are

$$L_i^{(r)} = \sum_{k=1}^{K-1} \|\mathbf{p}_i^{(r)}(k) - \mathbf{p}_i^{(r)}(k-1)\|_2, \quad L_{\text{mean}}^{(r)} = \frac{1}{N} \sum_{i=1}^N L_i^{(r)}. \quad (47)$$

The reported group statistics follow the implementation:

$$\bar{m}_v = \frac{1}{n_v} \sum_{r \in \mathcal{R}_v} m^{(r)}, \quad s_v = \sqrt{\frac{1}{n_v} \sum_{r \in \mathcal{R}_v} (m^{(r)} - \bar{m}_v)^2}, \quad (48)$$

where $m^{(r)}$ is any run-level metric, $\bar{m}_v$ is the group mean, and s_v is the population-form standard deviation.

5.2. Simulation 1: Rule-Based and LLM Decision Comparison

Simulation 1 compares three high-level decision variants: a pure rule-based baseline (rule), a local LLM using qwen2.5:3b, and a local LLM using qwen3.5:4b. All 30 Simulation 1 runs were completed successfully. In addition, the recorded field `llm_fallback_used` remained zero across all 20 LLM runs, indicating that no back-end failure triggered a run-time fallback to the rule policy. Figure 3 shows a representative rule-baseline run.

Table 2 reports the ten-seed results for the standard corridor. All three decision variants reach $\bar{Y} = 1.000$ and $\bar{R}_{\text{goal}} = 1.000$. This result supports the main safety claim that the feasibility-projection operator makes both rule and LLM decisions executable under the same geometric constraints. The rule baseline gives a slightly lower mean terminal error (6.126 m) than both LLM variants (6.206 m), but this difference is small relative to the common mission-level success. The LLM variants instead exhibit a clearer structural effect: they maintain a larger average number of sub-swarms (1.522 versus 1.456) and more structural events (38.60 versus 30.80). This indicates that the LLM layer does not merely reproduce the rule policy; it actively sustains split-merge organization while remaining inside the feasibility-projected action set. In the standard corridor, where the rule policy is already highly tuned, the LLM therefore matches the mission outcome while introducing richer high-level reconfiguration behavior. The aggregated per-metric statistics and the paired-seed differences underlying this comparison are reported in Figures 4 and 5, respectively.

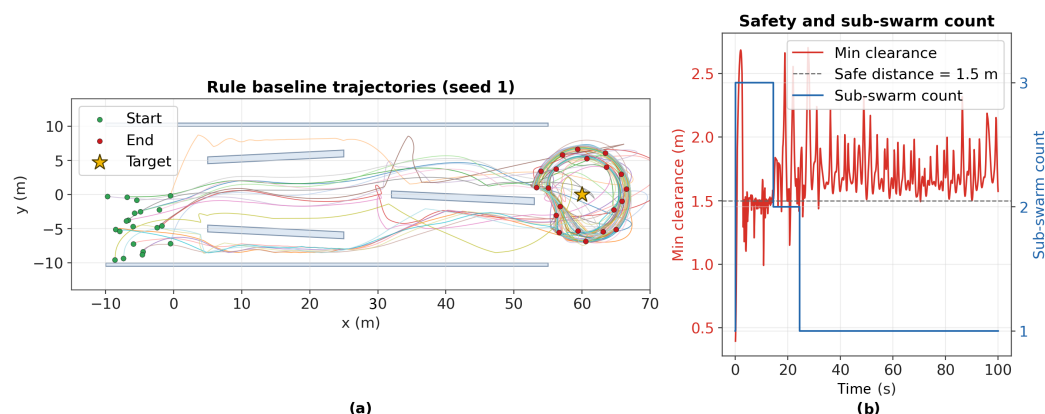


Figure 3. Rule baseline simulation result (seed 1). (a) Top-down UAV trajectories through the corridor: green dots mark the start positions, red dots the end positions, the yellow star the common target, the light-blue rectangles the obstacles, and the thin coloured curves the individual UAV paths; (b) time histories of the minimum clearance (red curve) and the sub-swarm count (blue curve), with the grey dashed line indicating the 1.5 m safe-distance threshold.

Table 2. Fine-grained results of Simulation 1 (mean $\pm$ std over 10 seeds).

Variant	$\bar{E}_{\text{mean}}$ (m)	$\bar{E}_{90}$ (m)	σ_E (m)	$\bar{N}_{\text{swarm}}$	$\bar{N}_{\text{event}}$	$\bar{R}_{\text{unsafe}}$	$\bar{L}_{\text{mean}}$ (m)
rule	6.126 ± 0.068	7.057 ± 0.066	0.916 ± 0.041	1.456 ± 0.109	30.80 ± 11.11	0.117 ± 0.025	224.59 ± 11.68
qwen2.5:3b	6.206 ± 0.050	7.167 ± 0.090	0.940 ± 0.054	1.522 ± 0.092	38.60 ± 9.26	0.126 ± 0.026	228.33 ± 10.92
qwen3.5:4b	6.206 ± 0.050	7.167 ± 0.090	0.940 ± 0.054	1.522 ± 0.092	38.60 ± 9.26	0.126 ± 0.026	228.33 ± 10.92

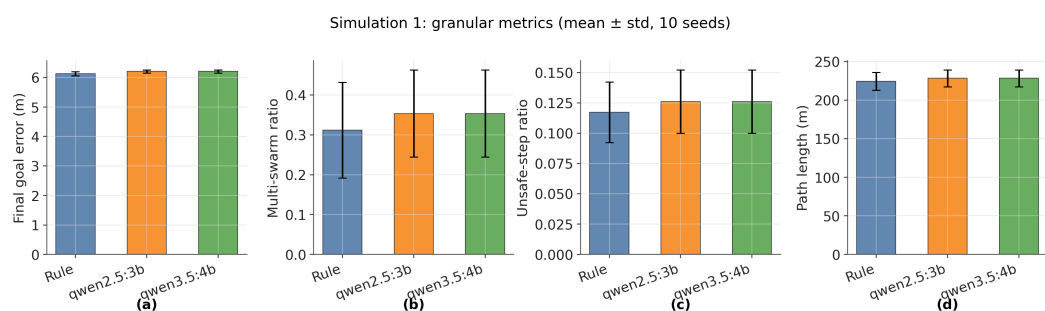


Figure 4. Fine-grained metrics for Simulation 1 aggregated over ten seeds. In every panel the bars from left to right denote the rule baseline, qwen2.5:3b, and qwen3.5:4b, and the error bars denote one standard deviation across the ten seeds: (a) final goal error; (b) multi-swarm ratio; (c) unsafe-step ratio; (d) path length.

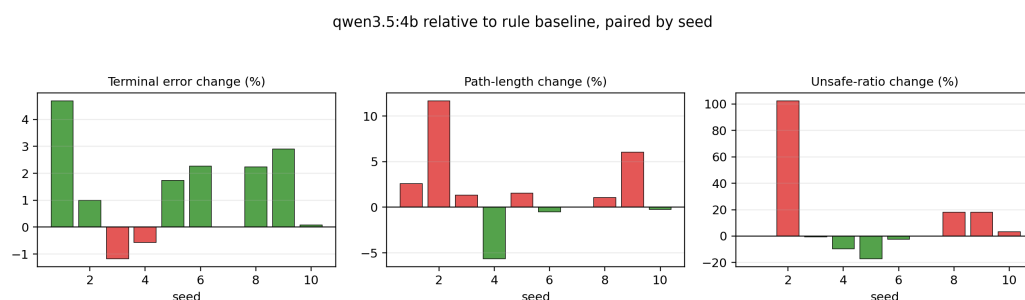


Figure 5. Paired-seed differences for Simulation 1. Bars report the percentage change for qwen3.5:4b relative to the rule baseline for matched seeds.

5.3. Simulation 2: Validation of the Risk-Control Mechanism

Simulation 2 is designed to validate the risk-control mechanism incorporated into the proposed navigation controller. In conventional swarm control models, the attraction between agents or between an agent and its target is typically computed regardless of whether the line of sight is obstructed by an obstacle. In contrast, the proposed controller introduces an occlusion attenuation factor in (30), which weakens attraction-related interactions when the connecting segment is blocked by an obstacle. Together with LiDAR-based obstacle repulsion and the geometric post-update safety projection, this forms the risk-control mechanism emphasized in this paper. The comparison is performed at the controller level. In the `safety_stack_on` setting, the full proposed risk-control controller is retained. In the `safety_stack_off` setting, the experimental code weakens the adaptive motion regulation layers around the same eight-force controller and relaxes the acceleration bounds to a very loose level. The validation proceeds in two complementary levels: a coarse whole-stack toggle (Table 3) first establishes that the risk-control mechanism as a whole is effective, and a fine-grained leave-one-out ablation (Table 4) then attributes the effect to its individual components. The two share the same ten seeds and the same `safety_stack_on` run as their common reference, and so the two tables are consistent rather than alternative views.

Table 3. Fine-grained ablation results of Simulation 2 (mean $\pm$ std over 10 seeds).

Variant	$\bar{E}_{\text{mean}}$ (m)	$\bar{L}_{\text{mean}}$ (m)	Path Stretch	$\bar{d}_{\text{clr,min}}$ (m)	$\bar{R}_{\text{unsafe}}$	$\bar{R}_{<0.75}$	$\bar{R}_{<0.30}$
<code>safety_stack_on</code>	6.126 ± 0.068	224.59 ± 11.68	3.459 ± 0.181	0.523 ± 0.427	0.117 ± 0.025	0.002 ± 0.002	0.001 ± 0.001
<code>safety_stack_off</code>	6.201 ± 0.108	319.19 ± 60.59	4.907 ± 0.925	0.106 ± 0.171	0.189 ± 0.146	0.140 ± 0.142	0.134 ± 0.140

From Table 3, it can be seen that the two settings give nearly the same terminal error (6.126 vs. 6.201 m). The main difference appears in the safety–efficiency profile. With `safety_stack_on`, the mean path length decreases from 319.19 to 224.59 m, which is a reduction of about 29.6%, and the path-stretch ratio decreases from 4.907 to 3.459. At the same time, the across-seed path-length standard deviation drops from 60.59 to 11.68, which indicates substantially better consistency.

The unsafe-step ratio decreases from 0.189 to 0.117, the near-wall ratio with $d_{\text{clr}} < 0.75$ m decreases from 0.140 to 0.002, and the extremely-close-wall ratio with $d_{\text{clr}} < 0.30$ m drops from 0.134 to nearly zero (specifically, 0.001). Therefore, the risk-control mechanism yields a tighter, more consistent, and safer passage through the corridor while maintaining almost the same mission-level accuracy. Figures 6 and 7 show the same tendency. Compared with `safety_stack_off`, `safety_stack_on` is consistently lower in path stretch, unsafe-step ratio, and near-wall/extremely-close-wall ratios, while maintaining almost identical terminal error.

The comparison above does not separate the contributions of individual components. To isolate them, we add a leave-one-out ablation over the same ten seeds (Table 4). The LiDAR-based repulsion is the primary avoidance term and is kept active throughout (removing it causes degeneration into trivial wall collisions). The two remaining components are toggled one at a time on top of the full controller: `w/o occ.` disables the occlusion attenuation ($\sigma_{\text{occ}} = 1$), and `w/o acc.` bound relaxes the acceleration saturation.

The two components act on distinct failure modes. The acceleration bound dominates efficiency and wall safety: relaxing it more than doubles the path length ($224.59 \rightarrow 484.55$ m, with variance $\pm 11.68 \rightarrow \pm 317.02$), inflates path stretch from 3.459 to 7.508, and raises the near-wall ratios $\bar{R}_{<0.75}$ ($0.002 \rightarrow 0.227$) and $\bar{R}_{<0.30}$ ($0.001 \rightarrow 0.220$)—reproducing most of the `safety_stack_off` degradation on its own. The occlusion attenuation has a smaller, separable effect on moderate-proximity risk: disabling it leaves path length

and the extremely-close-wall ratios essentially unchanged but raises the unsafe-step ratio $\bar{R}_{\text{unsafe}}$ from 0.117 to 0.138. Each component is thus non-redundant.

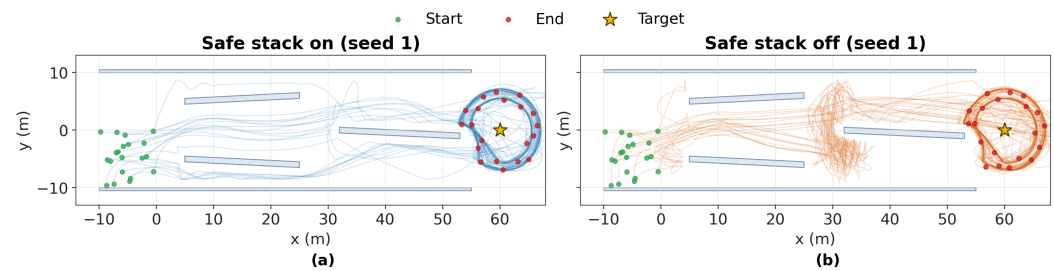


Figure 6. Risk-control ablation trajectory comparison (seed 1). (a) `safety_stack_on`: trajectories obtained with the full risk-control controller; (b) `safety_stack_off`: trajectories with the adaptive risk-control layers weakened and the acceleration bounds relaxed. Green dots mark the start positions, red dots the end positions, the yellow star the common target, and the light-blue rectangles the obstacles.

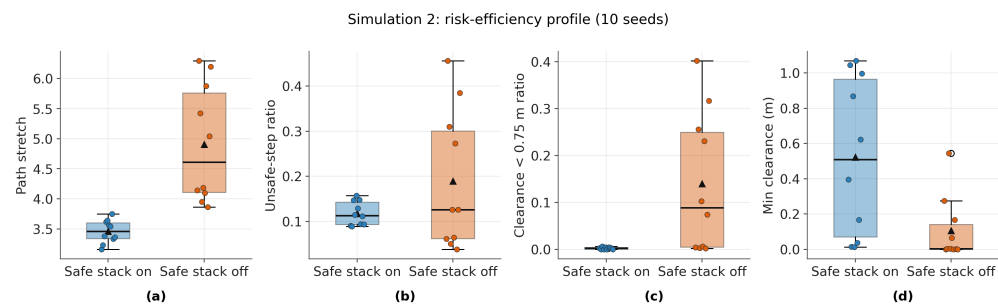


Figure 7. Risk-efficiency profile for Simulation 2 (ten seeds). Each panel contrasts `safety_stack_on` (blue) with `safety_stack_off` (orange): (a) path stretch; (b) unsafe-step ratio; (c) ratio of steps with clearance below 0.75 m; (d) minimum clearance. In each box the band is the median and the box spans the interquartile range, the triangle marks the mean, and the dots are the per-seed values.

Table 4. Leave-one-out component ablation of Simulation 2 (mean $\pm$ std over 10 seeds). LiDAR repulsion is kept active in all rows.

Variant	$\bar{E}_{\text{mean}}$ (m)	$\bar{L}_{\text{mean}}$ (m)	Path Stretch	$\bar{d}_{\text{clr,min}}$ (m)	$\bar{R}_{\text{unsafe}}$	$\bar{R}_{<0.75}$	$\bar{R}_{<0.30}$
Full safety stack	6.126 ± 0.068	224.59 ± 11.68	3.459 ± 0.181	0.523 ± 0.427	0.117 ± 0.025	0.002 ± 0.002	0.001 ± 0.001
w/o occ. attenuation	6.177 ± 0.095	223.88 ± 10.65	3.448 ± 0.169	0.526 ± 0.430	0.138 ± 0.034	0.002 ± 0.002	0.001 ± 0.001
w/o acc. bound	6.184 ± 0.128	484.55 ± 317.02	7.508 ± 4.975	0.262 ± 0.303	0.280 ± 0.290	0.227 ± 0.289	0.220 ± 0.282

5.4. Simulation 3: Navigation in More Complex Scenarios

Simulation 3 is designed to compare the rule and LLM decisions in more complex scenarios. Each scenario and decision mode is evaluated over ten random seeds, giving 80 successful Simulation 3 runs in total. All four scenarios share the same swarm initialization ($N = 20$ UAVs uniformly distributed in $[-10, 0]^2$ m) and channel outer walls at $y = \pm 10.4$ m. All runs use the same $K = 1000$ step horizon as Simulations 1–2. The LLM variant uses qwen3.5:4b in the same configuration as Simulation 1. The four scenarios differ in interior obstacle layout and goal placement:

- **default_corridor**: The standard scenario from Simulations 1–2. Two diagonal interior blockers ($x \in [5, 25]$, $y \approx \pm 5$ m) and one central separator ($x \in [32, 53]$, $y \approx 0$) create a two-branch passage; the goal is at (60, 0) m. This serves as the reference baseline.
- **narrow_alternating_gates**: Five single-sided gate walls alternate between upper and lower halves of the channel at $x \approx 12, 22, 32, 42$, and 52 m, each leaving only a sub-2.4 m gap on the opposite side. The swarm must make repeated, correctly directed split decisions; the goal is at (66, 0) m.
- **dense_cross_blocks**: Eight rectangular blocks are arranged at irregular positions, forming cross-shaped obstacles at $x \approx 15, 27, 39, 51$, and 62 m along the 65 m channel. Multiple simultaneous branching choices and frequent split–merge cycles are required; the goal is at (68, 0) m.
- **off_axis_goal**: Two pairs of symmetric lateral blockers and one central separator redirect the swarm, and the goal is placed at (70, 7.5) m (off the channel axis) to test goal-directed recovery when the required heading deviates from the corridor main axis.

To summarize cross-scenario robustness in a single quantity, the following composite scene score is introduced:

$$S_{\text{scene}} = 0.4Y + 0.3R_{\text{goal}} + 0.15(1 - R_{\text{unsafe}}) + 0.15 \text{clip}\left(\frac{d_{\text{clr,min}}}{r_s}, 0, 1\right), \quad (49)$$

where Y is the mission success rate, R_{goal} is the goal-reaching ratio, R_{unsafe} is the unsafe-step ratio, and $d_{\text{clr,min}}$ is the minimum clearance. A larger value of S_{scene} means better task completion, better risk control, and a wider clearance margin in the same scenario.

As presented in Table 5, the ten-seed evaluation shows that the LLM decision layer preserves full mission success in the easier **default_corridor** and **off_axis_goal** cases, matching the rule policy at the mission level while operating through language-generated high-level decisions. The strongest LLM benefit appears in the hardest **narrow_alternating_gates** scenario. Compared with the rule policy, the LLM increases mission success from 0.000 to 0.100, raises the goal-reaching ratio by 400%, reduces the mean terminal error by about 11%, and improves the composite scene score from 0.125 to 0.201. In the more cluttered **dense_cross_blocks** scenario, the rule policy is higher in mission success (0.600 versus 0.400), but the LLM remains close in terminal error (6.914 m versus 6.901 m), showing that the feasibility projection prevents semantic decisions from causing unstable behavior even in dense obstacle layouts. Overall, these ten-seed results support the role of the LLM as a semantic high-level reconfiguration module that preserves robustness in easy scenes and provides a clear recovery advantage in the most constrained alternating-gate case. Representative trajectories for the four scenarios are shown in Figure 8, and the corresponding scenario-level metric comparison is summarized in Figure 9.

Table 5. Simulation 3: rule-versus-LLM comparison after alignment to the first 1000 steps (means over ten seeds).

Scenario	Y_{rule}	Y_{llm}	$R_{\text{goal}}^{\text{rule}}$	$R_{\text{goal}}^{\text{llm}}$	$E_{\text{mean}}^{\text{rule}}$ (m)	$E_{\text{mean}}^{\text{llm}}$ (m)	$S_{\text{scene}}^{\text{rule}}$	$S_{\text{scene}}^{\text{llm}}$
default_corridor	1.000	1.000	1.000	1.000	6.126	6.184	0.885	0.880
narrow_alternating_gates	0.000	0.100	0.025	0.125	44.738	39.851	0.125	0.201
dense_cross_blocks	0.600	0.400	0.785	0.755	6.901	6.914	0.649	0.560
off_axis_goal	1.000	1.000	0.955	0.950	5.640	5.580	0.872	0.872

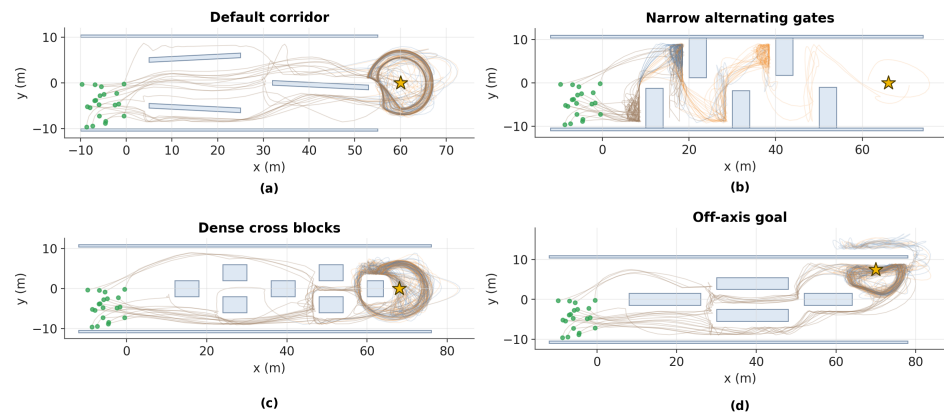


Figure 8. Representative scenario trajectories for Simulation 3. The panels illustrate the rule/LLM navigation behavior across the complex benchmark scenes without using the compressed multi-run montage. (a) Default corridor; (b) narrow alternating gates; (c) dense cross blocks; (d) off-axis goal. In each panel the green dots mark the start positions, the yellow star marks the scenario goal, the light-blue rectangles are the obstacles, and the thin brown curves are the UAV trajectories.

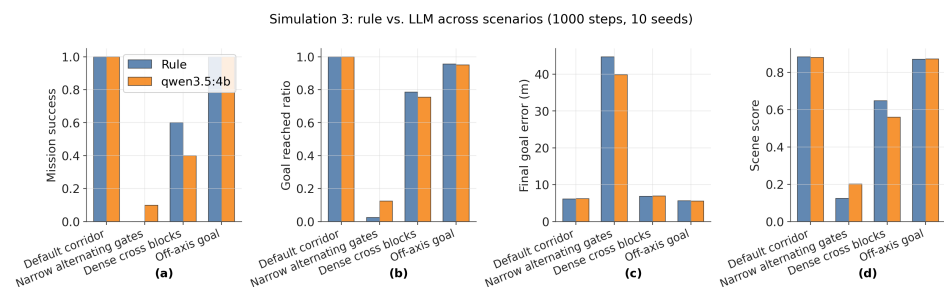


Figure 9. Scenario-level comparison for Simulation 3 using ten seeds per rule/LLM scenario pair. In every panel the blue bars denote the rule baseline and the orange bars the qwen3.5:4b LLM policy, grouped by the four benchmark scenarios: (a) mission success; (b) goal-reached ratio; (c) final goal error; (d) composite scene score.

6. Conclusions

This paper proposes a hierarchical embodied swarm framework for cooperative multi-UAV navigation in obstacle-rich environments, integrating corridor-driven structural decisions, feasibility-projected LLM assistance, and a multi-term force-based safety controller. Three key results are obtained from numerical simulations with $N = 20$ UAVs over $K = 1000$ steps, including a ten-seed decision-model comparison in Simulation 1 and a ten-seed complex-scenario comparison in Simulation 3. First, in the standard corridor scenario, all three decision variants (rule, qwen2.5:3b, and qwen3.5:4b) achieve 100% mission success and 100% goal-reaching ratio, confirming that the feasibility projection preserves correctness regardless of the decision source. Second, the risk-control mechanism reduces the mean path length by 29.6% (from 319.19 m to 224.59 m), cuts the unsafe-step ratio from 0.189 to 0.117, and reduces the near-wall ratio ($d_{\text{clr}} < 0.75$ m) from 0.140 to 0.002 while maintaining nearly identical terminal accuracy. Third, in the complex-scenario evaluation, the LLM-assisted variant preserves feasibility and mission-level robustness across all scenarios; it matches the rule policy in the easier scenes, and in the hardest narrow_alternating_gates case, it raises mission success from 0.000 to 0.100, increases the goal-reaching ratio from 0.025 to 0.125, and reduces mean terminal error from 44.738 m to 39.851 m.

Beyond the empirical evaluation, the smooth conservative core of the controller is backed by a formal guarantee: Theorem 1 establishes, via a mechanical-energy Lyapunov function, that under a known (possibly time-varying) cruise velocity whose acceleration

is supplied as feedforward, the formation tracking error stays uniformly bounded by the initial energy, with the proof detailed in Appendix B.

This paper investigates the problem of dynamic self-organization and safe navigation in hierarchical embodied swarms and proposes a hierarchical framework that integrates passability-driven structural decisions with force-based motion control. By introducing a feasibility projection operator, both rule-based and LLM-based high-level decisions are executed under unified hard safety constraints. The multi-seed simulations show that in simple scenarios, the LLM-based path preserves the same coarse mission reliability as the rule path; in challenging scenarios, it supplies a feasible semantic reconfiguration mechanism that improves recovery-related indicators under the most constrained gate layout. The proposed risk-control mechanism enhances passage safety and trajectory quality, preventing overly aggressive decisions from the LLM. Future work will focus on two aspects. First, more robust split–merge triggering mechanisms will be developed and terminal convergence will be improved. Second, a small-scale multi-quadrotor platform will be used to validate the proposed framework under real-world sensing, communication, and actuation uncertainties, with a focus on split–merge triggering, corridor negotiation, and robustness. In addition, event-triggered low-level path-following control with explicit output constraints will be investigated to further reduce control updates and strengthen formal safety guarantees during narrow-passage flight.

Author Contributions: Conceptualization, L.W. and C.W.; Methodology, L.W. and C.W.; Software, L.W.; Validation, L.W.; Formal analysis, L.W.; Investigation, L.W.; Resources, L.W.; Data curation, L.W.; Writing—original draft, L.W.; Writing—review & editing, L.W. and C.W.; Visualization, L.W.; Supervision, C.W.; Project administration, C.W.; Funding acquisition, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under grant numbers U24B20156, 62350048, U2541218, T2121003 and 62533006.

Data Availability Statement: Data will be made available on request.

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

Appendix A. Three-Dimensional Perspective Views of Simulation 1

To improve the readability of the standard-corridor experiment without changing the underlying evaluation protocol, Figure A1 provides two representative perspective renderings from the same planar simulation log used in Simulation 1 (qwen3.5:4b, seed 1). The UAVs are shown at the simulated fixed flight altitude of $z = 10$ m.

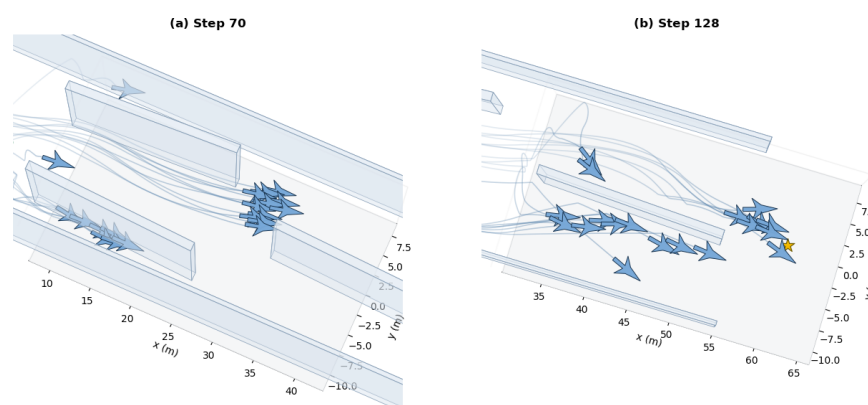


Figure A1. Perspective visualization of two representative stages in Simulation 1. (a): the swarm negotiates the split corridor while maintaining two branches. (b): after passing the central separator,

the sub-swarms begin to regroup toward the common goal. The blue arrows denote the individual UAVs and their instantaneous heading directions, the yellow star marks the common goal, and the light-grey slabs are the corridor walls and obstacles. This figure is generated from the same planar simulation data analyzed in the main text and is included only to improve scene interpretability.

Appendix B. Lyapunov Error-Boundedness Analysis of the Smooth Controller Core

This appendix proves that, within a frozen-structure smooth window $\mathcal{W} = [t_0, t_1)$, the conservative core of the controller (38) is energy-dissipating and keeps the formation error uniformly bounded by the initial energy while the formation cruises at a known, possibly time-varying velocity, the known cruise acceleration being compensated by feed-forward. Of the eight force terms, the first seven form this smooth core; the eighth, the LiDAR repulsion $f_{\text{obs},i}$, is discontinuous and is set aside from the outset, together with occlusion attenuation, saturation, and split-merge switching. The result is thus a boundedness (safety) certificate, not a proof of global convergence to $\mathbf{p}_g$. Throughout, t is continuous time on $\mathcal{W}$; i, j index UAVs, $d_{ij} := \|\mathbf{p}_i - \mathbf{p}_j\|$; and the remaining symbols follow the main text. The four reductions below render the closed loop analytically tractable—outside such a window, several terms are non-differentiable, and no smooth energy function exists. With exact model knowledge, the inverse model $\mathbf{u}_i = m_i(\mathbf{a}_i^{\text{cmd}} + c_d \mathbf{v}_i)$ cancels the plant damping exactly ($\mathbf{a}_i = \mathbf{a}_i^{\text{cmd}}$), and so the closed loop reduces to the double integrator:

$$\dot{\mathbf{p}}_i = \mathbf{v}_i, \quad \dot{\mathbf{v}}_i = \mathbf{a}_i^{\text{cmd}}. \quad (\text{A1})$$

We analyze a single scenario: the formation translates rigidly at a common, possibly time-varying cruise velocity $\mathbf{v}_s(t)$, such that the slot moves as $\mathbf{p}_i^*(t) = \mathbf{p}_i^*(t_0) + \int_{t_0}^t \mathbf{v}_s(\tau) d\tau$ with reference velocity $\mathbf{v}_i^* = \mathbf{v}_s(t)$, and the cruise acceleration $\dot{\mathbf{v}}_s(t)$ is *known*. Define the formation error $\mathbf{e}_i := \mathbf{p}_i - \mathbf{p}_i^*(t)$ and, consequently, the error velocity $\dot{\mathbf{e}}_i := \mathbf{v}_i - \mathbf{v}_s$. The position feedback restores $\mathbf{e}_i$, the velocity feedback damps $\dot{\mathbf{e}}_i$, and the known $\dot{\mathbf{v}}_s$ is supplied to every agent as acceleration feedforward.

Potential construction. Under these conditions, the nominal core is described by a nonnegative C^1 potential on $\mathcal{W}$. Let $\mathcal{E}_{\text{sep}}$ and $\mathcal{E}_{\text{ir}}$ denote the fixed active unordered pair sets for global separation and intra-sub-swarm repulsion, and define

$$\begin{aligned} \tilde{V}_p(\mathbf{p}) = & \underbrace{\sum_i \frac{k_p}{2} \|\mathbf{p}_i - \mathbf{p}_i^*\|^2}_{V_{\text{att}}} + \underbrace{\sum_i \frac{k_{\text{path}}}{2} \|\mathbf{p}_{i,xy} - \mathbf{c}_s\|^2}_{V_{\text{path}}} + \tilde{V}_{\text{aux}} \\ & + \underbrace{\sum_{(i,j) \in \mathcal{E}_{\text{sep}}} k_{\text{sep}} \log \frac{r_{\text{sep}}}{d_{ij}}}_{V_{\text{sep}}} + \underbrace{\sum_{(i,j) \in \mathcal{E}_{\text{ir}}} k_{\text{ir}} \log \frac{d_{\text{ir}}}{d_{ij}}}_{V_{\text{ir}}}, \end{aligned} \quad (\text{A2})$$

where $\tilde{V}_{\text{aux}} \geq 0$ collects any restoring intra-swarm/follower/cohesion components implemented in fixed-reference or symmetrized conservative form. The logarithmic primitives are dictated by the implemented radial repulsion $k(\mathbf{p}_i - \mathbf{p}_j)/d_{ij}^2$: writing $k > 0$ for the corresponding gain (k_{sep} or k_{ir}) and R for the corresponding radius (r_{sep} or d_{ir}), on each active set,

$$-\nabla_{\mathbf{p}_i} \left(k \log \frac{R}{d_{ij}} \right) = k \frac{\mathbf{p}_i - \mathbf{p}_j}{d_{ij}^2}.$$

By construction, the entire position part of $\mathbf{a}_i^{\text{cmd}}$ equals $-\nabla_{\mathbf{p}_i} \tilde{V}_p$, the velocity feedback damps the error velocity $\dot{\mathbf{e}}_i$, and the known cruise acceleration $\dot{\mathbf{v}}_s$ enters as feedforward; thus, the nominal smooth core reads

$$\begin{aligned} \mathbf{a}_i^{\text{cmd}} &= \mathbf{a}_i^{\text{core}} := -\nabla_{\mathbf{p}_i} \tilde{V}_p - D_i(\mathbf{v}_i - \mathbf{v}_s) + \dot{\mathbf{v}}_s, \\ D_i &= \begin{cases} k_v I, & i \text{ subleader,} \\ (k_v + k_{\ell_v}) I, & i \text{ follower,} \end{cases} \quad D_i = D_i^\top \succ 0, \end{aligned} \quad (\text{A3})$$

where D_i is the positive-definite local damping matrix; the common cruise cancels in the follower coupling $k_{\ell_v}(\mathbf{v}_{\ell_s} - \mathbf{v}_i) = k_{\ell_v}(\dot{\mathbf{e}}_{\ell_s} - \dot{\mathbf{e}}_i)$, which is therefore unaffected.

Formation-error boundedness. Because all slots share the cruise velocity $\mathbf{v}_s$, the inter-slot offsets are constant, and so $\tilde{V}_p$ is a time-invariant function of $\mathbf{e} = (\mathbf{e}_i)$ with $V_{\text{att}} = \sum_i \frac{k_p}{2} \|\mathbf{e}_i\|^2$. Differentiating the error $\mathbf{e}_i$ twice along (A1) and recalling $\dot{\mathbf{e}}_i = \mathbf{v}_i - \mathbf{v}_s$ gives the error acceleration

$$\ddot{\mathbf{e}}_i = \dot{\mathbf{v}}_i - \dot{\mathbf{v}}_s = \mathbf{a}_i^{\text{cmd}} - \dot{\mathbf{v}}_s. \quad (\text{A4})$$

Substituting the command (A3), the feedforward $+\dot{\mathbf{v}}_s$ cancels the reference acceleration $-\dot{\mathbf{v}}_s$ exactly, leaving the error dynamics

$$\ddot{\mathbf{e}}_i = -\nabla_{\mathbf{e}_i} \tilde{V}_p - D_i \dot{\mathbf{e}}_i, \quad (\text{A5})$$

which is *identical* to the static (non-cruising) loop: the moving-formation problem is reduced to the fixed-formation problem with no residual drive. Take the mechanical-energy candidate $V = \frac{1}{2} \sum_i \|\dot{\mathbf{e}}_i\|^2 + \tilde{V}_p(\mathbf{e})$, bounded below and radially unbounded in $\mathbf{e}$ through V_{att} .

Proof of Theorem 1. By differentiating $V = \frac{1}{2} \sum_i \|\dot{\mathbf{e}}_i\|^2 + \tilde{V}_p(\mathbf{e})$ along (A5) and noting that the potential-gradient and kinetic cross terms cancel because the feedforward removed the drive, we can obtain

$$\dot{V} = \sum_i \dot{\mathbf{e}}_i^\top \ddot{\mathbf{e}}_i + \sum_i (\nabla_{\mathbf{e}_i} \tilde{V}_p)^\top \dot{\mathbf{e}}_i = -\sum_i \dot{\mathbf{e}}_i^\top D_i \dot{\mathbf{e}}_i \leq 0, \quad (\text{A6})$$

the last inequality by $D_i \succ 0$. Hence, V is non-increasing, and so $V(t) \leq V(t_0)$ on $\mathcal{W}$. Since $\tilde{V}_p \geq V_{\text{att}} = \sum_i \frac{k_p}{2} \|\mathbf{e}_i\|^2$ and $\frac{1}{2} \sum_i \|\dot{\mathbf{e}}_i\|^2 \geq 0$, for each i ,

$$\frac{k_p}{2} \|\mathbf{e}_i(t)\|^2 \leq V(t) \leq V(t_0), \text{ i.e.,}$$

$$\|\mathbf{e}_i(t)\| \leq \sqrt{\frac{2V(t_0)}{k_p}}, \quad \forall t \in \mathcal{W}, \forall i, \quad (\text{A7})$$

which is the bound (40) asserted in Theorem 1. Both the error position $\mathbf{e}_i$ and the error velocity $\dot{\mathbf{e}}_i = \mathbf{v}_i - \mathbf{v}_s$ are therefore bounded for all $t \in \mathcal{W}$. $\square$

Corollary A1 (Velocity and nominal error convergence via LaSalle). *Because the feedforward renders the error loop (A5) autonomous and V is non-increasing with compact, positively invariant sublevel sets, LaSalle's invariance principle applies whenever the frozen smooth regime persists ($t_1 \rightarrow \infty$): every trajectory converges to the largest invariant set contained in $\{\dot{V} = 0\} = \{\dot{\mathbf{e}}_i = \mathbf{0}, \forall i\}$. On this set, $\dot{\mathbf{e}}_i \equiv \mathbf{0}$ forces $\ddot{\mathbf{e}}_i \equiv \mathbf{0}$; hence, $\nabla_{\mathbf{e}_i} \tilde{V}_p = \mathbf{0}$, and so the invariant set is*

$$\mathcal{M} = \{(\mathbf{e}, \dot{\mathbf{e}}) : \dot{\mathbf{e}} = \mathbf{0}, \nabla_{\mathbf{e}} \tilde{V}_p(\mathbf{e}) = \mathbf{0}\}. \quad (\text{A8})$$

Consequently, the error velocity converges to zero, $\dot{\mathbf{e}}_i \rightarrow \mathbf{0}$, and the configuration converges to the set of critical points of $\tilde{V}_p$ (force equilibria where attraction balances the active separation/repulsion/cohesion/centering terms). In the nominal frozen smooth window, if the desired formation is the isolated critical point of $\tilde{V}_p$ inside the invariant sublevel set reached by the trajec-

tory, then $\mathcal{M} = \{(\mathbf{0}, \mathbf{0})\}$ locally and LaSalle's invariance principle further yields $\mathbf{e}_i(t) \rightarrow \mathbf{0}$ for all i .

Appendix C. Simulation Parameter Summary

Table A1 consolidates the key parameter values used in all simulations, organized by functional category. Force gains were co-tuned so that formation tracking converges stably at the nominal simulation speed while inter-agent spacing is maintained above d_{safe} under LiDAR repulsion. The split-merge cooldown $T_{\text{cd}} = 10$ s prevents rapid oscillation between structural states. The follower reassignment improvement threshold $\Delta_{\text{min}} = 2$ m requires a clear geometric benefit before triggering a transfer.

Table A1. Key simulation parameters grouped by functional category.

Parameter	Value	Description
Dynamics and sensing		
$N, \Delta t$	20, 0.1 s	Swarm size; integration step
m_i	5 kg	UAV mass
R_L, R_t	20 m, 14 m	LiDAR range; repulsion trigger radius
$\Delta\theta$	1°	LiDAR angular resolution ($n_\theta = 360$ rays)
a_{xy}^{max}	2.6 m/s ²	Acceleration saturation
Force-synthesis gains		
k_p, k_v	0.7, 0.8	Attraction position/velocity gains
k_{rep}, k_e	9.0, 48.0	LiDAR repulsion base gain; emergency gain
k_b, γ	0.9, 1.2	Short-range boost factor and exponent
$k_{\text{sep}}, k_{\text{intra}}, k_{\text{ir}}$	1.1, 1.8, 1.6	Separation; intra-swarm cohesion; intra-swarm repulsion gains
$k_{\text{path}}, k_\ell, k_{\ell v}$	0.75, 1.6, 0.7	Corridor centering; follower-subleader coupling gains
k_{coh}	0.9	Centroid cohesion gain
$d_{\text{safe}}, \sigma_{\text{occ}}$	1.5 m, 0.2	Safety distance; occlusion attenuation factor
High-level structural decisions		
T_{cd}	10 s	Split-merge cooldown
$r_{\text{in}}, r_{\text{out}}$	8 m, 11 m	Encircle enter/exit thresholds
$r_{\text{merge}}, d_{\text{merge}}$	12 m, 8 m	Near-goal and inter-swarm merge thresholds
Follower reassignment		
w_ℓ, w_c	1.0, 0.5	Subleader- and waypoint-distance cost weights
$\Delta_{\text{min}}, \tau_{\text{ra}}$	2.0 m, 2 s	Minimum improvement threshold; per-follower cooldown
$N_{\text{move}}^{\text{max}}$	3	Maximum number of transfers per planner update
LLM interface		
T_{llm}	50 steps (5 s)	LLM invocation interval
Decoding temperature	0 (greedy)	Deterministic LLM decoding setting
Max tokens	64	Maximum output length
Timeout	120 s	Per-call inference timeout

References

- Skorobogatov, G.; Barrado, C.; Salamí, E. Multiple UAV systems: A survey. *Unmanned Syst.* **2020**, *8*, 149–169. [\[CrossRef\]](#)
- Javed, S.; Hassan, A.; Ahmad, R.; Ahmed, W.; Ahmed, R.; Saadat, A.; Guizani, M. State-of-the-art and future research challenges in UAV swarms. *IEEE Internet Things J.* **2024**, *11*, 19023–19045. [\[CrossRef\]](#)
- Wang, Y. Reinforcement Learning-Based Methods for Cooperative Control of UAVs: Challenges and Perspectives. *Guid. Navig. Control* **2025**, *5*, 435–438. [\[CrossRef\]](#)
- Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. *ACM Siggraph Comput. Graph.* **1987**, *21*, 25–34. [\[CrossRef\]](#)
- Vicsek, T.; Czirók, A.; Ben-Jacob, E.; Cohen, I.; Shochet, O. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **1995**, *75*, 1226. [\[CrossRef\]](#)
- Shafiq, M.; Ali, Z.A.; Israr, A.; Alkhamash, E.H.; Hadjouni, M. A multi-colony social learning approach for the self-organization of a swarm of UAVs. *Drones* **2022**, *6*, 104. [\[CrossRef\]](#)
- Chen, B.; Hu, J.; Ghosh, B.K. Finite-time tracking control of heterogeneous multi-AUV systems with partial measurements and intermittent communication. *Sci. China Inf. Sci.* **2024**, *67*, 152202. [\[CrossRef\]](#)
- Xu, B.; Bai, G.; Liu, T.; Fang, Y.; Zhang, Y.a.; Tao, J. An improved swarm model with informed agents to prevent swarm-splitting. *Chaos Solitons Fractals* **2023**, *169*, 113296. [\[CrossRef\]](#)

9. Zhang, C.; Ji, L.; Yang, S.; Guo, X.; Li, H. Distributed optimal consensus control for multiagent systems based on event-triggered and prioritized experience replay strategies. *Sci. China Inf. Sci.* **2025**, *68*, 112206. [\[CrossRef\]](#)
10. Zhu, H.; Juhl, J.; Ferranti, L.; Alonso-Mora, J. Distributed multi-robot formation splitting and merging in dynamic environments. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2019; pp. 9080–9086. [\[CrossRef\]](#)
11. Zhu, K.; Han, B.; Zhang, T. Multi-UAV distributed collaborative coverage for target search using heuristic strategy. *Guid. Navig. Control* **2021**, *1*, 2150002. [\[CrossRef\]](#)
12. Zhang, X.; Ding, W.; Wang, Y.; Luo, Y.; Zhang, Z.; Xiao, J. Bio-inspired self-organized fission-fusion control algorithm for UAV swarm. *Aerospace* **2022**, *9*, 714. [\[CrossRef\]](#)
13. Wang, N.; Jia, W.; Wu, H.; Wang, Y. Event-triggered self-organizing swarm control of distributed unmanned surface vehicles. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 3431–3445. [\[CrossRef\]](#)
14. Sendra-Arranz, R.; Gutierrez, A.; Christensen, A.L. Evolution of transferable and self-organized communication modules for solving multiple swarm robotics tasks. *IEEE Trans. Cybern.* **2026**, *56*, 595–608. [\[CrossRef\]](#)
15. Li, J.; Fang, Y.; Cheng, H.; Wang, Z.; Wu, Z.; Zeng, M. Large-scale fixed-wing UAV swarm system control with collision avoidance and formation maneuver. *IEEE Syst. J.* **2023**, *17*, 744–755. [\[CrossRef\]](#)
16. Zhu, W.; Ouz, S.; Heinrich, M.K.; Allwright, M.; Wahby, M.; Christensen, A.L.; Garone, E.; Dorigo, M. Self-organizing nervous systems for robot swarms. *Sci. Robot.* **2024**, *9*, ead15161. [\[CrossRef\]](#) [\[PubMed\]](#)
17. de Groot, O.; Ferranti, L.; Gavrila, D.M.; Alonso-Mora, J. Topology-driven parallel trajectory optimization in dynamic environments. *IEEE Trans. Robot.* **2025**, *41*, 110–126. [\[CrossRef\]](#)
18. Zhang, G.; Li, Z.; Li, J.; Huang, J. Robust Event-Triggered Path Following Control for a Rotor-Assisted Vehicle With Output Constraints: Theory and Experiment. *IEEE Trans. Ind. Electron.* **2026**, *73*, 10594–10604. [\[CrossRef\]](#)
19. Wang, M.; Niu, Y.; Wang, B.; Zhang, W.; Wang, C. A survey on learning motion planning and control for mobile robots: Toward embodied intelligence. *IEEE Trans. Neural Netw. Learn. Syst.* **2026**, Early Access. [\[CrossRef\]](#)
20. Feng, Z.; Xue, R.; Yuan, L.; Yu, Y.; Ding, N.; Liu, M.; Gao, B.; Sun, J.; Zheng, X.; Wang, G. Multi-agent embodied AI: Advances and future directions. *arXiv* **2025**. [\[CrossRef\]](#)
21. Liu, S.; Zhang, H.; Qi, Y.; Wang, P.; Zhang, Y.; Wu, Q. AerialVLN: Vision-and-Language Navigation for UAVs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2023; pp. 15384–15394. [\[CrossRef\]](#)
22. Guo, M.; Wu, M.; He, J.; Li, S.; Li, H.; Tao, C. BEDI: A comprehensive benchmark for evaluating embodied agents on UAVs. *ISPRS J. Photogramm. Remote Sens.* **2026**, *232*, 910–936. [\[CrossRef\]](#)
23. Wang, X.; Yang, D.; Liao, Y.; Zheng, W.; Dai, B.; Wu, W.; Li, H.; Liu, S. UAV-Flow Colosseo: A real-world benchmark for flying-on-a-word UAV imitation learning. *arXiv* **2025**, arXiv:2505.15725. [\[CrossRef\]](#)
24. Agal, S.; Odedra, N.D. Decentralized reinforcement learning for scalable embodied intelligence in robotic swarms. *Embodied Intell. Robot.* **2025**, *X*, 1–16. [\[CrossRef\]](#)
25. Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Gopalakrishnan, K.; Hausman, K.; et al. Do as I can, not as I say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning (CoRL 2022)*, Auckland, New Zealand, 16 August 2022. [\[CrossRef\]](#)
26. Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of the 6th Conference on Robot Learning (CoRL 2022)*, Auckland, New Zealand, 16 August 2022. [\[CrossRef\]](#)
27. Kannan, S.S.; Venkatesh, V.L.N.; Min, B. SMART-LLM: Smart multi-agent robot task planning using large language models. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Abu Dhabi, UAE, 2024; pp. 12140–12147. [\[CrossRef\]](#)
28. Yu, H.; Wang, C.; Niu, Y.; Wu, L. MUTP-LLM: Empowering Multi-UAV Task Planning with Large Language Models. *Guid. Navig. Control* **2025**, *5*, 477–489. [\[CrossRef\]](#)
29. Zhu, M.; Pang, T.; Gao, M.; Xu, H.; Yu, G.; Xia, F. A novel UAV swarm decision-making system under unified task and spatial view. In *Proceedings of the International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC)*, Guilin, China, 8–10 August 2025. [\[CrossRef\]](#)
30. Hai, X.; Qiu, H.; Wen, C.; Feng, Q. A novel distributed situation awareness consensus approach for UAV swarm systems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 14706–14717. [\[CrossRef\]](#)
31. Wang, Z.; Li, J.; Li, J.; Liu, C. A decentralized decision-making algorithm of UAV swarm with information fusion strategy. *Expert Syst. Appl.* **2024**, *237*, 121444. [\[CrossRef\]](#)
32. Wei, Z.; Wei, R. An information aggregation decision making method for UAV swarm intelligence system based on joint communication and proximal strategy. *Expert Syst. Appl.* **2026**, *298*, 129617. [\[CrossRef\]](#)
33. Shahid, S.; Zhen, Z.; Javaid, U.; Wen, L. Offense-defense distributed decision making for swarm vs. swarm confrontation while attacking the aircraft carriers. *Drones* **2022**, *6*, 271. [\[CrossRef\]](#)

34. Xia, W.; Zhou, Z.; Jiang, W.; Zhang, Y. Dynamic UAV swarm confrontation: An imitation based on mobile adaptive networks. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 7183–7202. [[CrossRef](#)]
35. Jiang, Q.; Yan, Y.; Dai, Y.; Yang, Z.; Cao, H.; Wang, B.; Ma, X. Autonomous task planning of intelligent unmanned aerial vehicle swarm based on deep deterministic policy gradient. *Drones* **2025**, *9*, 272. [[CrossRef](#)]
36. Zheng, Z.; Wei, C.; Duan, H. UAV swarm air combat maneuver decision-making method based on multi-agent reinforcement learning and transferring. *Sci. China Inf. Sci.* **2024**, *67*, 180204. [[CrossRef](#)]
37. Chen, W.; Hai, X.; Hu, Y.; Feng, Q.; Wang, Z. Hierarchical decision-making framework for multi-UAV task assignment via enhanced pigeon-inspired optimization. *Guid. Navig. Control* **2023**, *3*, 2350028. [[CrossRef](#)]
38. Roy, D.; Maitra, M.; Bhattacharya, S. Exploration of multiple unknown areas by swarm of robots utilizing virtual-region-based splitting and merging technique. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 3459–3470. [[CrossRef](#)]
39. Gazi, V.; Passino, K.M. Stability analysis of swarms. *IEEE Trans. Autom. Control* **2003**, *48*, 692–697. [[CrossRef](#)]
40. Olfati-Saber, R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control* **2006**, *51*, 401–420. [[CrossRef](#)]
41. Tanner, H.G.; Jadbabaie, A.; Pappas, G.J. Flocking in fixed and switching networks. *IEEE Trans. Autom. Control* **2007**, *52*, 863–868. [[CrossRef](#)]

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.