MP-CDE: Motion Planning for Autonomous Quadrotor Swarm in Communication-Denied Environments with Active Perception

Xingxun Liu, *Student Membership*, Yaonan Wang, *Membership*, Zhiqiang Miao*, *Member, IEEE*, Wei He, *Fellow IEEE*, Xiangke Wang, *Senior Member IEEE*

***Abstract*—Overcoming communication constraints remains a critical challenge for autonomous quadrotor swarms in practical deployments. Existing state-of-the-art solutions often assume idealized communication conditions, and their performance degrades in the presence of communication disturbances. This paper presents MP-CDE, a novel motion planning framework for autonomous quadrotor swarm navigation in communication-denied environments. Our key innovation lies in systematically exploiting the inherent physical coupling between aerial mobility and visual perception capabilities. Through dynamic yaw angle optimization, the framework enables individual quadrotors to achieve dual objectives of maximizing environmental perception and maintaining operational safety. Comprehensive validation through simulations and real-world experiments demonstrates that MP-CDE enables robust swarm navigation even under complete communication blackouts. The code and video are available at https://xxliu-hnu.github.io/mp-cde/.**

***Index Terms*—Communication denial, quadrotors, multi-robot systems, motion planning, collision avoidance, swarm robotics.**

I. Introduction

QUADROTOR swarms are increasingly deployed for applications such as last-mile delivery [1], large-scale cooperative aerial surveying [2], and formation-flight demonstrations [3]. These deployments favor decentralized architectures for their scalability and robustness in unstructured settings; however, such designs typically presuppose sustained, high-bandwidth inter-agent links. This modeling choice creates a theory–practice gap: in real deployments, communication-denied or degraded conditions—due to electromagnetic interference, structural occlusions can expose vulnerabilities in conventional decentralized swarm strategies [4].

Xingxun Liu, Yaonan Wang, and Zhiqiang Miao are with the School of Artificial Intelligence and Robotics, Hunan University, Changsha 410082, China, and also with the National Engineering Research Center for Robot Visual Perception and Control, Changsha 410012, China (e-mail: liuxingxun@hnu.edu.cn; yaonan@hnu.edu.cn; miaozhiqiang@hnu.edu.cn).

Xiangke Wang is with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: xkwang@nudt.edu.cn).

Wei He is with the School of Automation and Institute of Artificial Intelligence, Beijing Information Science and Technology University, Beijing 102206, China, and also with the School of Intelligence Science and Technology, University of Science and Technology, Beijing 100083, China (e-mail: weihe@ieee.org).

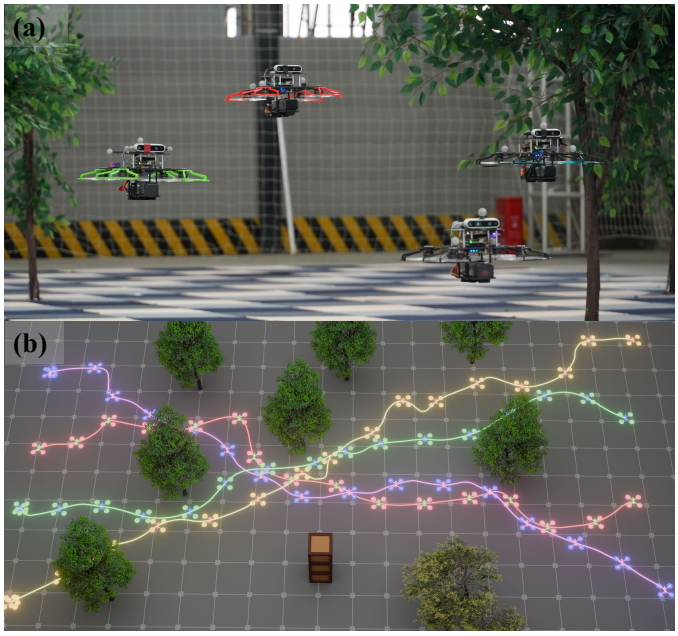

Fig. 1. The four quadrotors use onboard sensors to achieve safe navigation in a complex communication-denied environment. The motion capture system is only used to visualize the position of the quadrotors.

Specifically, in communication-denied environments, commonly adopted decentralized swarm methods [5], [6] encounter two critical limitations: (1) intermittent neighbor-position updates critically impair conventional collision avoidance models that rely on real-time location data; and (2) navigation capabilities are severely constrained by the limited horizontal field of view (FoV) of commercial-grade sensors in the absence of comprehensive environment perception. For instance, widely adopted depth sensors such as the Intel RealSense D435i offer only an 84° horizontal FoV, resulting in substantial perception blind spots that hinder timely environment mapping and dynamic obstacle detection.

Although some decentralized methods [7]–[10] have employed sensor-based prediction to estimate agent trajectories for collision-free planning, two critical limitations remain in practical deployments. First, the requirement for additional or high-performance sensors with extended perception capabilities poses significant challenges for small-scale quadrotors,

as payload and cost constraints limit their feasibility. Second, the assumption of obstacle-free environments or reliance on external positioning systems, such as GPS or motion capture, rarely holds in complex, cluttered, and communication-denied environments, where both perception and coordination must operate under stringent sensing and information constraints.

To bridge this gap, we propose **MP-CDE** (**M**otion **P**lanning for autonomous quadrotor swarm in **C**ommunication-**D**enied **E**nvironments), a motion planning approach designed for autonomous quadrotor swarms operating in communication-denied environments. Unlike prior work that relies on explicit communication, our framework exploits the physical coupling between quadrotor motion and sensor perception range. By actively optimizing yaw angles to maximize environmental observability, each quadrotor strategically expands its environment perception while maintaining essential safety margins. To detect other agents within the swarm, we employ a learning-based detection algorithm. Next, a lightweight yet effective matching algorithm associates detected targets with trackers to predict their trajectories. Finally, a multi-objective optimization approach is applied to resolve conflicts within the swarm, ensuring smooth and coordinated operation.

To the best of our knowledge, this method is the first to achieve fully autonomous swarm planning in communication-denied environments. Compared with the existing state-of-the-art methods, the proposed method can eliminate conflicts within the swarm by actively observing the trajectories of other quadrotors in the swarm as a substitute for the real trajectory. We conduct comprehensive experiments in both simulation and the real world to validate our method. The contributions of this paper are:

1) A safety-metric-driven yaw planning strategy that explicitly reasons about viewpoint availability under communication denial, together with well-defined mode transitions between exploring, observing, and tracking behaviours.
2) An adaptive, perception-aware prediction pipeline that fuses RGB-D detections with covariance-matched Kalman filtering and supplies consistency certificates for the downstream MINCO optimizer.
3) A reproducible open-source system implementation that tabulates all planning and sensing parameters, publicly releases flight logs and real-world trajectory reconstructions, and demonstrates sustained operation across diverse cluttered environments.

II. Related Work

We systematically categorize existing approaches through a communication-constraint lens, establishing three distinct operational regimes: (1) Ideal communication with perfect state synchronization; (2) Delayed communication with bounded-time message delivery; (3) Communication-denied environments requiring sensor-driven autonomy. We use the term *synchronous* to denote planners that require simultaneous trajectory publication or shared timing guarantees, whereas *asynchronous* methods permit independent replanning as long as bounded communication delay assumptions hold. Communication-denied methods, by contrast, must rely on on-board sensing alone and cannot assume any neighbor-to-neighbor message exchange. Table II summarizes representative works according to these criteria, highlighting the associated sensor suite and safety assumptions. In particular, the communication-denied category remains comparatively sparse, with most prior efforts either constraining to two-dimensional motion [17] or requiring external markers and specialized hardware such as ultraviolet LEDs [19].

A. Ideal Communication

State-of-the-art methods under ideal communication conditions typically assume perfect inter-agent state sharing through dedicated networks, enabling collision-free trajectory generation through centralized coordination principles.

DFCC [11] is a novel formation planning methodology that combines differential flatness technology with control barrier functions, achieving a superior balance between cooperative control and obstacle avoidance. A formation feedback controller produces trajectories, which require inter-robot communication to acquire the other quadrotor's current positions and intended trajectories. EGO-Swarm [5], an extension of EGO-Planner [20], is a gradient-based, decentralized, and asynchronous system that enables autonomous navigation of quadrotor swarms in unknown, obstacle-rich environments using only on-board resources. Conflict handling relies on inter-robot communication to fetch future segments of peer trajectories; MADER [12], an extension of FASTER [21], enables motion planning for quadrotors in dynamic, dense environments by representing obstacles and quadrotors as convex polyhedra, which are then decomposed into an optimization problem to be solved using collision detection operations. Primitive-Swarm [13] utilizes a time-optimal motion primitive library and occupancy information calculated offline to transform the time-consuming optimization problem into a selection problem with linear complexity. This approach enables the system to fully explore non-convex, discontinuous three-dimensional safe spaces filled with numerous obstacles and robots, while running ultra-large-scale swarm simulations in real time.

While enabling provably safe coordination, these methods exhibit critical fragility when facing real-world communication impairments.

B. Delayed communication

RMADER [14] is an extension of MADER designed to handle communication delays. It introduces a delay detection method, a two-step broadcast trajectory publishing scheme, and a novel trajectory storage and inspection mechanism to manage communication delays effectively.

HDSM [15] is a synchronous planning method that simultaneously plans two trajectories - one for a known environment and one for an unknown environment. At the beginning of each iteration, if the agent has not received the trajectory of at least one nearby agent, it will continue to execute its last generated trajectory to cope with communication delays.

Dream [16] is a decentralized real-time multi-robot trajectory planning algorithm in which each agent computes discrete

TABLE I
RELATED WORK COMPARISON ON MULTI-ROBOT MOTION PLANNING REGARDING COMMUNICATION CONDITION, DISTRIBUTED/ASYNCHRONOUS PLANNING, COMMUNICATION RELIANCE, AND OBSTACLE PRESENCE

Comm. Condition	Method	Dist.	Async.	Comm. Needed	Sensor Type
Ideal	DFCC [11]	Yes	Yes	Yes	No†
	EGO-Swarm [5]	Yes	Yes	Yes	Camera
	MADER [12]	Yes	No	Yes	Camera
	Primitive-Swarm [13]	Yes	No	Yes	Camera
Delayed	RMADER [14]	Yes	No	Yes	Camera
	HDSM [15]	Yes	No	Yes	No†
	Dream [16]	Yes	Yes	Yes	No†
No/Denied	Zhu et al. [8]	Yes	Yes	No	No†
	Hybrid-MRCA [17]	Yes	No	No	No†
	Pan et al. [18]	Yes	No	No	No†
	RBL-Controller [19]	Yes	No	No	Camera

† The location of the obstacle is known or there are no obstacles in the environment.

separating hyperplane trajectories (DSHTs) [22] concerning other agents. DSHTs are then incorporated into the decision-making process to prevent collisions, enabling asynchronous planning and ensuring safe operation under imperfect communication.

C. Denied communication

Zhu et al. [8] proposed a decentralized communication-free motion planner by combining an RNN-based interaction-aware trajectory predictor with MPC. While their method achieves performance comparable to centralized planners in homogeneous teams, it assumes all robots follow identical planning policies and exhibits limited robustness when interacting with heterogeneous agents or in unseen environments.

Hybrid-MRCA [17] is a decentralized sensor-level collision avoidance method that maps raw sensor inputs directly to agent velocities, avoiding the need for frequent inter-agent communication. However, it is limited to two-dimensional scenarios. Extending it to three-dimensional environments significantly increases computational complexity due to higher degrees of freedom and a larger state space, making it unsuitable for real-time 3D applications.

Pan et al. [18] proposed a real-time communication-free decentralized algorithm for navigating the robot to its desired goal by control barrier functions, which model and control the onboard sensing behavior to keep neighbors in the limited FoV for position estimation. However, this method has only been tested in open environments and cannot be applied to environments filled with obstacles.

RBL-Controller [19] is a distributed swarming algorithm capable of operating in complex environments without inter-agent communication. However, this vision-based approach requires specialized hardware support, including grayscale cameras equipped with optical filters and blinking ultraviolet LEDs for agent identification.

The goal of MP-CDE is to achieve safety planning in unknown and complex environments by relying only on quadrotor sensors without any external positioning devices or communication devices. By explicitly coupling yaw planning, perception-driven safety metrics, and adaptive neighbor prediction, MP-CDE addresses the remaining gap in the literature: a fully vision-based, communication-denied swarm planner with quantified safety guarantees.

III. SYSTEM OVERVIEW AND PROBLEM FORMULATION

Consider a communication-denied environment $\mathcal{E} \subset \mathbb{R}^3$ containing a set of static obstacles $\mathcal{O} = \{\mathbf{o}_m\}_{m=1}^{M}$. In this environment, a swarm of N quadrotors operates autonomously without inter-agent communication. For each quadrotor i, at time t, its position and yaw orientation are denoted by $\mathbf{p}_i(t)$ and $\psi_i(t)$, respectively. The primary objective of the proposed framework is to simultaneously generate dynamically feasible position trajectories $\mathbf{p}_i(t)$ and yaw trajectories $\Omega_i(t)$ for all quadrotors, while guaranteeing collision avoidance with both the static obstacles and other agents in the swarm. It can be formulated as:

$$\begin{cases} \operatorname{dist}(\mathbf{p}_i(t), \mathcal{O}) \geq r_{\text{robot}}, & \forall i = 1, \ldots, N \\ \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| \geq 2 * r_{\text{robot}}, & \forall i, j = 1, \ldots, N,\ i \neq j, \\ \|\dot{\mathbf{p}}_i(t)\| \leq v_{\max}, & \forall i = 1, \ldots, N, \\ \|\ddot{\mathbf{p}}_i(t)\| \leq a_{\max}, & \forall i = 1, \ldots, N. \end{cases}$$

where $\operatorname{dist}(\mathbf{p}, \mathcal{O})$ denotes the shortest Euclidean distance from point $\mathbf{p}$ to the obstacle set $\mathcal{O}$. r_{robot} is the radius of the quadrotor. $v_{\max}$ and $a_{\max}$ denote the maximum velocity and acceleration of the quadrotor, respectively.

Each quadrotor is equipped with onboard perception, localization, and planning modules that rely solely on its own sensor data, thus enabling fully autonomous operation without global positioning systems or inter-agent communication.

As illustrated in Fig. **??**, the MP-CDE system employs identical hardware and software configurations across all quadrotors. Each platform integrates an RGB-D camera and an inertial measurement unit (IMU).

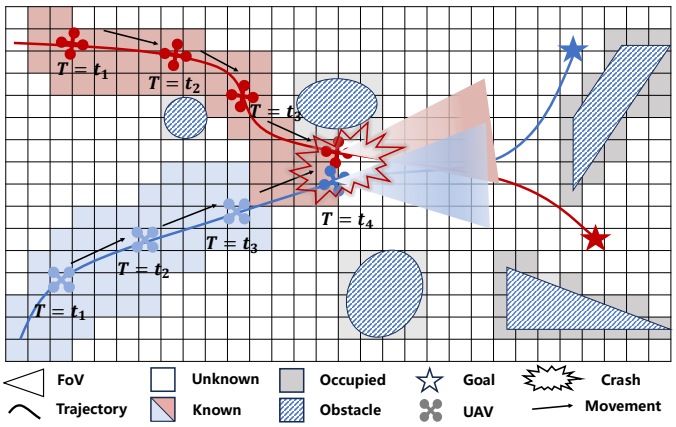

Fig. 2. Collisions occur in communication-denied environments. Due to the limited sensing range, when two quadrotors follow approximately parallel trajectories, they may remain undetected to each other, potentially leading to a collision.

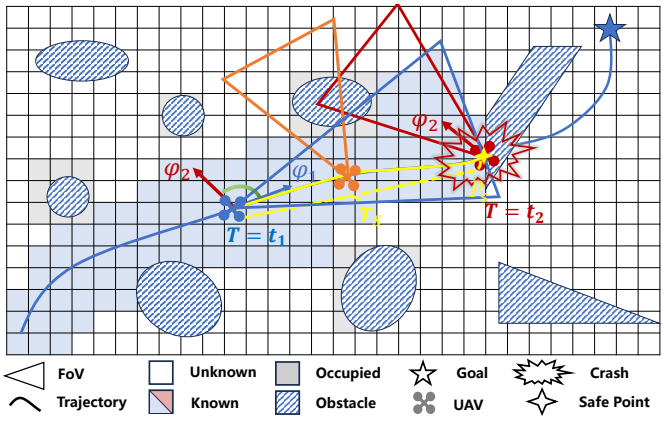

Fig. 3. Collisions occur during active observation. When the quadrotor actively perceives, it risks hitting obstacles in these unobserved regions, where φ_1 and φ_2 denote its yaw angles at times t_1 and t_2, respectively.

To mitigate the impact of limited sensing range, a hierarchical yaw planning framework (Section IV-A) proactively selects observation angles to maximize the likelihood of target detection, while strictly enforcing safety constraints. Target information is processed through a lightweight matching-prediction module (Section IV-B), which associates current observations with predicted trajectories. In the presence of intra-swarm conflicts, the trajectory planning module (Section IV-C) performs real-time, collision-free path replanning.

The framework is fully decentralized, requiring neither GNSS nor explicit inter-agent communication. Reliable operation in denied environments is achieved through onboard sensor fusion and predictive coordination, thereby enabling each quadrotor to make autonomous decisions based on locally available information.

IV. Methodology

Conventional decentralized swarms exchange their local plans to coordinate collision avoidance; in the absence of such communication channels, state uncertainty escalates and the limited FoV of the onboard cameras quickly leads to blind spots, as illustrated in Fig. 2. Our framework therefore proceeds in three tightly coupled stages: **Active environmental perception** actively steers the camera to enlarge the verified free space while guaranteeing that braking remains feasible; **Lightweight target matching and tracking** fuses all detections, even if intermittent, into consistent neighbor predictions that remain valid during temporary occlusion; **Collision-free trajectory generation** continuously replans using the predicted trajectories and safety margins so that the swarm remains separated despite sensing and communication limitations.

A. Active environmental perception

To enhance environmental perception capability, existing approaches such as [23], [24] exploit the differential flatness property of quadrotors [25] by separately planning yaw angles after trajectory generation. However, this decoupled planning scheme will lead to inconsistency between the sensor direction and the quadrotor's movement direction, increasing the risk of collision, as shown in Fig. 3. In this paper, a safety-constrained adaptive method is proposed to maximize environment perception while guaranteeing collision-free operations through dynamic yaw angle planning.

We introduce two quantities that bind perception geometry to the motion plan. The safe point P_s denotes the furthest waypoint along the current trajectory that lies within the depth-based occupancy map confirmed as free space, and the associated safe time τ_s is the travel time needed to reach P_s when following the commanded speed profile. For any look-ahead parameter $\tau \in [0, \tau_s]$ we denote by $P(\tau)$ the position that the quadrotor would occupy after advancing τ seconds along the spline trajectory.

These quantities define the safety margin

$$S(\tau) = \min\left(\frac{d_{\text{free}}(P(\tau))}{v_{\max}},\ \tau_s - \tau\right), \tag{1}$$

where $d_{\text{free}}(P(\tau))$ is the Euclidean signed-distance (ESDF) value of the map at $P(\tau)$. The first term guarantees that enough free distance remains to brake before reaching unobserved space, while the second term ensures that the segment ahead has been temporally validated by the perception pipeline. We require $S(\tau) > 0$ for safe operation; equality indicates that the vehicle is about to leave the verified corridor.

To avoid unnecessary stops, we introduce a switching threshold

$$\tau_t = \epsilon \frac{\pi}{\dot{\psi}_{\max}} \tag{2}$$

with $\epsilon = 1.2$ in all experiments. The quantity $\pi/\dot{\psi}_{\max}$ corresponds to the worst-case half-turn needed to redirect the camera from its current heading to the opposite side; a full 2π rotation is redundant because the yaw planner never requests consecutive rotations in the same direction without revisiting the previously cleared sector. The slack ϵ covers sensing latency and controller tracking errors measured during calibration. States with $S(\tau) \geq \tau_t + 0.1\,\text{s}$ enter the *exploring*

or *observing* modes, whereas $S(\tau) \leq \tau_t$ forces the *tracking* mode. The 0.1 s hysteresis between these bounds eliminates chattering when the metric oscillates near the threshold.

As the quadrotor navigates along its trajectory, the proposed framework continuously evaluates $S(\tau)$ to decide whether to prioritise exploration, observation, or trajectory tracking.The mode transition logic is formally implemented in Alg. 1.

1) Exploring Mode: During the safety-guaranteed phase, the generation of the yaw path is modeled as a graph search problem seeking a sequence of yaw angles $\Omega_\psi := \{(\psi_i)_{i=0:N}\}$, inspired by [26]. We first uniformly sample each node and construct the graph

$$G = (V, E),$$

$$V = \bigcup_{i=1}^{n} \left\{ \psi_{i,j} \;\middle|\; j = 1, 2, \ldots, m \right\}$$

$$E = \bigcup_{i=1}^{n-1} \left\{ e_{j,k} \;\middle|\; \begin{array}{l} j \in \{1, 2, \ldots, m\}, \\ k \in \{1, 2, \ldots, m\}, \\ \|\psi_{i,j} - \psi_{i+1,k}\| < \dot{\psi}_{\max} \cdot t_\psi \end{array} \right\}$$

In this construction $i \in \{0, \ldots, n\}$ indexes discrete look-ahead steps separated by the sampling period t_ψ, whereas j (and k) enumerate the m candidate yaw angles per step. Edges therefore only connect adjacent time layers that obey the yaw-rate constraint, which yields a sparse lattice solvable in $\mathcal{O}(nm \log(nm))$ time via Dijkstra or A*.

The edge cost function for graph search is formulated as a weighted combination of two complementary metrics:

- **Exploration Metric (E_m)**: This metric encapsulates the exploration reward, quantifying coverage efficiency. It drives the system towards unknown areas through real-time information entropy accumulation.
- **Visibility Metric (V_m)**: This metric boosts yaw configurations that keep previously detected neighbors inside the FoV by forecasting their predicted trajectories and assigning higher rewards to viewpoints that sustain mutual observability.

For efficient exploration quantification, we propose a lightweight coverage estimator based on yaw trajectory analysis. The core mechanism maintains a sliding window $\mathcal{W}_i = \{\psi_{i-w}, ..., \psi_i\}$ of recent yaw angles, where w defines the observation memory horizon. For each candidate yaw we render an angular sector that is clipped by the point-cloud based ESDF to account for obstacle occlusions; this is implemented via 72 discrete rays and therefore avoids the cubic complexity of dense ray-casting. The coverage area S_i is computed as the union of these occlusion-aware sectors swept along $\mathcal{W}_i$. The incremental exploration gain is then formulated as:

$$E_m^i = S_i - S_{i-1} = \mu\left(\mathcal{A}(\mathcal{W}_i) \setminus \mathcal{A}(\mathcal{W}_{i-1})\right) \tag{3}$$

where $\mu(\cdot)$ denotes area measure and $\mathcal{A}(\cdot)$ represents the FoV-covered region.

In order to reduce the risk of collision within the swarm, it is necessary to optimize Ω_ψ to improve the target observation coverage. Leveraging the trajectory prediction framework detailed in Sec. IV-B, the observation feasibility evaluation model can be established by analyzing the predicted trajectory $\mathbf{p}_k(t)$ of the target quadrotor k, the observer quadrotor's position $\mathbf{p}_i$ and yaw angle ψ_{ij}. The visibility reward is then formulated as:

$$V_m^i = \sum_{k=1}^{s} \sum_{t \in \mathcal{T}_i} w(t)\, \mathbb{I}(\mathbf{p}_k(t) \in \mathrm{FoV}(\psi_{i,j}, \mathbf{p}_i)) \frac{1}{d_{k,i}(t) + \epsilon} \tag{4}$$

where $\mathcal{T}_i$ denotes a discrete set of future time samples spanning $[0, \tau_s]$, $w(t)$ is a triangular weighting kernel that emphasises near-term visibility, $\mathbb{I}(\cdot)$ activates the reward only when quadrotor k lies inside the FoV, and $d_{k,i}(t)$ is the Euclidean distance between the predicted target position and the observer. The small constant ϵ prevents division by zero and caps the reward when the neighbor is extremely close.

These two metrics are normalised to $[0, 1]$ and merged into an edge reward $R_i = \alpha E_m^i + \beta V_m^i$ that we wish to maximise. In practice we assign the negative reward as the edge cost used by Dijkstra's algorithm:

$$C_i = -(\alpha E_m^i + \beta V_m^i) \tag{5}$$

where α and β are positive tuning coefficients (set to 0.6 and 0.4 respectively in all experiments). After completing the construction of the graph we solve a shortest-path problem whose optimal solution therefore corresponds to the maximum-reward yaw sequence.

2) Observing Mode: When the safety budget is generous ($S(\tau) \geq \tau_t + 0.1\,\mathrm{s}$) the planner queries the tracker for the set $\mathcal{V}$ of neighbors whose last visual update occurred within the hold time $t_{\text{hold}} = 0.6$ s. If $\mathcal{V}$ contains all actively tracked neighbors, the yaw is steered to maximise joint visibility through a distance-weighted synthesis:

$$\psi_{\text{observe}} = \frac{\sum_{q \in \mathcal{V}} \lambda_q \arctan 2(y_q - y_i, x_q - x_i)}{\sum_{q \in \mathcal{V}} \lambda_q}, \tag{6}$$

where $\lambda_q = 1/(d_{i,q} + \epsilon)$ emphasises closer agents. If at least one active neighbor has not been re-detected within t_{hold}, the planner switches back to the exploring mode so that the camera deliberately scans the occluded sector before the prediction covariance exceeds the gating limit.

3) Tracking Mode: When $S(\tau)$ is less than the τ_t, it means that only a small part of the future trajectory is safe. In order to avoid collision with obstacles, the direction of the quadrotor is adjusted to the forward direction of the trajectory. This setting can effectively ensure safety while exploring the environment.

Algorithm 1 Yaw Angle Planning Strategy

$\mathcal{A} \leftarrow$ active EKF trackers (Sec. IV-B)
$\mathcal{V} \leftarrow \{q \in \mathcal{A} \mid t - t_q^{\text{last}} < t_{\text{hold}}\}$ ▷ neighbors recently observed
$S(\tau), \tau_s \leftarrow$ Evaluate safety metric (Eq. 1) along current trajectory
if $S(\tau) \geq \tau_t + 0.1\,\text{s}$ **then**
  if $|\mathcal{V}| = |\mathcal{A}|$ **then**
    $\psi \leftarrow \textsc{ObserveYaw}(\mathcal{V})$ ▷ Eq. 4 weighting
  else
    $\psi \leftarrow \textsc{ExploreYaw}(\mathcal{A} \setminus \mathcal{V})$ ▷ graph search in Sec. IV-A
  end if
else
  $\psi \leftarrow \textsc{TrackYaw}$(trajectory heading)
end if
if $S(0) \leq \tau_t \vee \min_q d_{i,q}(0) < C_{\mathbf{s}}$ **then**
  MINCO-$\textsc{Replan}(\psi, \mathfrak{T}_{\text{MINCO}})$ ▷ Sec. IV-C
end if

After each yaw decision, the predicted neighbor states are propagated forward and fed to the MINCO optimizer described in Sec. IV-C. Neighbors that remain unseen for more than t_{hold} are treated as dynamic obstacles with an inflated clearance radius of $1.2\,C_{\mathbf{s}}$ until they are re-acquired, which prevents the planner from relying on stale predictions. This mechanism, together with the proactive exploration sweep, explicitly handles scenarios in which two UAVs temporarily lose mutual visibility: either the yaw planner rotates back to recover the target, or the inflated virtual obstacle triggers a MINCO replanning step that separates the trajectories.

B. Lightweight target matching and tracking

The neighbor-perception module runs at 25 Hz and consists of three stages.[1] First, a YOLOv11 detector executing on the onboard Jetson Orin NX processes 416×416 RGB frames and outputs axis-aligned bounding boxes with an average latency of 45 ms. Depth values are simultaneously obtained from the aligned Intel RealSense D435i stream at 30 Hz. Second, each detection $d_t = (u_t, v_t, d_t^z)$ is back-projected into the camera frame using the calibrated intrinsic matrix $\mathbf{K}$:

$$\mathbf{p}_t^c = d_t^z\, \mathbf{K}^{-1}[u_t, v_t, 1]^\top. \tag{7}$$

Finally, the 3-D point is transformed to the world frame through the IMU–camera extrinsic calibration $(\mathbf{R}_{cw}, \mathbf{p}_{cw})$ and the current body pose $(\mathbf{R}_{bw}, \mathbf{p}_{bw})$:

$$\mathbf{p}_t^w = \mathbf{R}_{bw}(\mathbf{R}_{cw}\mathbf{p}_t^c + \mathbf{p}_{cw}) + \mathbf{p}_{bw}. \tag{8}$$

Depth outliers caused by reflective surfaces are rejected via a spatial median filter applied over a 3×3 neighborhood.

State representation and dynamics: Each tracked neighbor q is represented by the state vector $\mathbf{x}_k^q = [\mathbf{p}_k^{q\top}, \mathbf{v}_k^{q\top}]^\top \in \mathbb{R}^6$ at discrete time index k. We intentionally adopt a constant-velocity (CV) model because it offers the best trade-off between fidelity and computational cost for the short prediction horizons ($\tau_s \leq 1.2$ s) considered here; higher-order models produced negligible accuracy gains in our experiments while doubling the parameter dimension. The CV transition is

$$\mathbf{x}_{k+1}^q = \mathbf{F}(\Delta t_k)\,\mathbf{x}_k^q + \mathbf{w}_k^q, \qquad \mathbf{F}(\Delta t) = \begin{bmatrix} \mathbf{I}_3 & \Delta t\,\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}, \tag{9}$$

where Δt_k is the elapsed time since the previous update (typically 40–50 ms) and $\mathbf{w}_k^q \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ denotes process noise. The available measurement is the 3-D position $\mathbf{z}_k^q = \mathbf{p}_t^w$ from (8), yielding the observation model

$$\mathbf{z}_k^q = \mathbf{H}\,\mathbf{x}_k^q + \mathbf{v}_k^q, \qquad \mathbf{H} = [\mathbf{I}_3\ \mathbf{0}_3], \tag{10}$$

with measurement noise $\mathbf{v}_k^q \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. Velocities are therefore inferred rather than directly measured, addressing the reviewer's concern about inconsistent notation in the original manuscript.

Adaptive covariance update: Following the covariance-matching strategy of [27], the empirical innovation $\boldsymbol{\gamma}_k^q = \mathbf{z}_k^q - \mathbf{H}\,\hat{\mathbf{x}}_{k|k-1}^q$ is used to update the process noise estimate:

$$\mathbf{C}_{\gamma,k} = \boldsymbol{\gamma}_k^q \boldsymbol{\gamma}_k^{q\top}, \qquad \hat{\mathbf{Q}}_k = \mathbf{C}_{\gamma,k} - \mathbf{F}(\Delta t_{k-1})\,\mathbf{P}_{k-1}\,\mathbf{F}^\top(\Delta t_{k-1}) - \mathbf{R}_k. \tag{11}$$

To mitigate numerical fluctuations we employ an exponential moving average $\mathbf{Q}_k = \alpha\hat{\mathbf{Q}}_k + (1-\alpha)\mathbf{Q}_{k-1}$ with $\alpha = 0.3$, which we found to balance responsiveness and robustness.

Data association and occlusion handling: Data association relies on the Hungarian algorithm with Mahalanobis costs. A $\chi^2_{3,0.975} = 7.38$ gating threshold accommodates sensor noise. Tracks without matched detections for $t_{\text{hold}} = 0.6$ s keep propagating via (9) but their covariance is inflated linearly, which is reflected in the yaw-planning weight $w(t)$ in (4). If the missed-detection duration exceeds 1.0 s the track is pruned. Conversely, re-detected neighbors immediately re-enter the active set $\mathcal{A}$ used in Algorithm 1.

Prediction output: The planner queries each track for a short-term prediction

$$\mathbf{p}^q(t) = \hat{\mathbf{p}}_k^q + \hat{\mathbf{v}}_k^q(t - t_k), \qquad t \in [t_k, t_k + \tau_s], \tag{12}$$

which is consistent with the CV assumption and sufficient for the MINCO horizon in Sec. IV-C. Sec. V-B2 reports the sensitivity of this prediction to the smoothing coefficient α and to the gating threshold.

C. Collision-Free Trajectory Generation

We adopt the minimum control effort polynomial trajectory $\mathfrak{T}_{\text{MINCO}}$ [28] as the representation of trajectories for quadrotors, which can be deformed in time and space through optimization. The $\mathfrak{T}_{\text{MINCO}}$ is written as:

$$\begin{aligned} \mathfrak{T}_{\text{MINCO}} = \ & \{p(t) : [0, T] \mapsto \mathbb{R}^m \mid \mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T}), \\ & \mathbf{q} \in \mathbb{R}^{m\times(M-1)}, \mathbf{T} \in \mathbb{R}^M_{>0}\} \end{aligned} \tag{13}$$

Here, $p(t)$ is an m-dimensional piece-wise polynomial trajectory with M pieces and degree $N = 2\sigma - 1$, where σ is the order of the integrator chain. $\mathbf{q} = (q_1, \ldots, q_{M-1})$ represents the intermediate points, and $\mathbf{T} = (T_1, \ldots, T_M)^T$ is the time vector. The trajectory $\mathfrak{T}_{\text{MINCO}}$ is parameterized only by $\mathbf{q}$ and $\mathbf{T}$. The polynomial coefficient $\mathbf{c} = (\mathbf{c}_{T1}, \ldots, \mathbf{c}_{TM})^T \in$

[1]The source code and annotated flight logs are available at https://xxliu-hnu.github.io/mp-cde/.

$\mathbb{R}^{2M\sigma\times m}$ can be obtained by the mapping $\mathcal{M}(\mathbf{q},\mathbf{T})$, where $\mathbf{c}_k \in \mathbb{R}^{2\sigma\times m}$ is the coefficient matrix of the k-th piece. The k-th piece of $p(t)$ is represented by

$$p_k(t) = \mathbf{c}_k^T \beta(t) \tag{14}$$

where $\beta(t) = \left(1, t, \ldots, t^N\right)^T$ is the natural basis, and $t \in [0, T_k]$.

In our work, the trajectory generation is modeled as a multi-objective optimization problem, which includes the requirements of smoothness, safety, aggressiveness, dynamical feasibility, and small formation error. The optimization problem for formation flight is formulated as:

$$\min\left(w_c f_c + w_t f_t + w_I f_I + w_s f_s\right) \tag{15}$$

where w_c, w_t, w_I, and w_s are the positive weights. The cost function consists of: control energy cost f_c, time cost f_t, and penalty cost of soft constraints f_I, safety cost f_s. The safety cost $f_\mathbf{s}$ is defined as:

$$f_\mathbf{s} = \sum_{i=0}^{\kappa} \max\left\{C_\mathbf{s}^2 - \|\mathbf{E}^{1/2}(p(\tau) - p_i(\tau))\|, 0\right\}^3 \tag{16}$$

where $\mathbf{E} := (1, 1, 1/c)$ with $c > 1$ transforms a Euclidean distance into an ellipsoidal distance with the minor axes at the z axis to relieve the downwash risk from rotors. $C_\mathbf{s}$, named swarm clearance, is the minimum safety clearance between two quadrotors. $p_i(\tau)$ is the trajectory of the i-th detected quadrotors, obtained in Section IV-B. The definitions of other cost functions can be found in [29].

V. Experiment and Evaluations

This section presents evaluations of our system in both simulated and real-world environments. Thorough quantitative analyses are conducted to assess the performance and advantages of the proposed system.

A. Implementation Details

The proposed method is first tested and validated in the Robot Operating System (ROS) simulation environment and then in real-world experiments.

For the simulation test, we used a laptop with an Intel i9 12900H CPU running at about 2.8 GHz. Apart from simulations, we designed real-world experiments to validate the performance and robustness of our method. The main experimental setup of the formation system is shown in Fig. 4. We built a physical experimental platform for the experimental requirements, based on a Q250 airframe, equipped with an Intel RealSense D435i depth camera and an NVIDIA Jetson ORIN NX running Ubuntu 20.04 as the onboard computer. The quadrotor is equipped with an NxtPX4 V2 controller executing the PX4 flight stack. The overall quadrotor platform weighs only 0.80 kg and measures 180× 180× 75 mm. The entire hardware platform is lightweight, with a thrust-to-weight ratio of 4.2.

TABLE II
KEY PLANNING AND PERCEPTION PARAMETERS

Parameter	Description	Value
$\dot{\psi}_{\max}$	Maximum yaw rate	2.0 rad/s
t_ψ	Yaw sampling period	0.1 s
n	Graph horizon steps	12
α, β	Yaw search weights	$0.6, 0.4$
t_{hold}	Visibility hold time	0.6 s
t_{drop}	Track pruning time	1.0 s
$C_\mathbf{s}$	Swarm clearance radius	0.7 m
α_{EMA}	Covariance smoothing factor	0.3

Details of our hardware platform have been made publicly available as open-source[2]. Table II summarises the key planning and perception parameters shared across all simulated and real-world experiments.

B. Simulation Experiment

1) Evaluation of Active Perception: To evaluate the proposed active environmental perception, ablation and comparative experiments were conducted in simulated environments with varying obstacle densities. The experimental setup is illustrated in Fig. 5. System performance was evaluated using the observed area S_{obs} and the task success rate R_s as metrics. These two metrics inherently conflict. Overly aggressive exploration may increase S_{obs} but lead to collisions, whereas overly conservative strategies preserve R_s at the cost of reduced coverage. The impact of mode switching was assessed by selectively disabling either the exploring mode or the tracking mode.

The results of experiments are shown in Table III. In comparative experiments against state-of-the-art (SOTA) approaches [26], the proposed method consistently achieves higher S_{obs} and R_s, especially in cluttered environments. In the ablation study, removing the exploring mode substantially reduces S_{obs} and limits perception coverage, whereas disabling the tracking mode significantly reduces R_s and compromises navigation robustness.

TABLE III
ABLATION STUDY OF THE ACTIVE PERCEPTION

Obs. density	Method	$S_{obs}(m^2)$
Condition 1	Zhou's [26]	142.7
	Ours (w/o exploring mode)	137.1
	Ours (w/o tracking mode)	/
	Ours	**167.5**
Condition 2	Zhou's [26]	132.4
	Ours (w/o exploring mode)	122.4
	Ours (w/o tracking mode)	/
	Ours	**145.9**
Condition 3	Zhou's [26]	105.7
	Ours (w/o exploring mode)	94.6
	Ours (w/o tracking mode)	/
	Ours	**123.5**

[2]https://github.com/HNU-CAT/HNU-Quadrotor

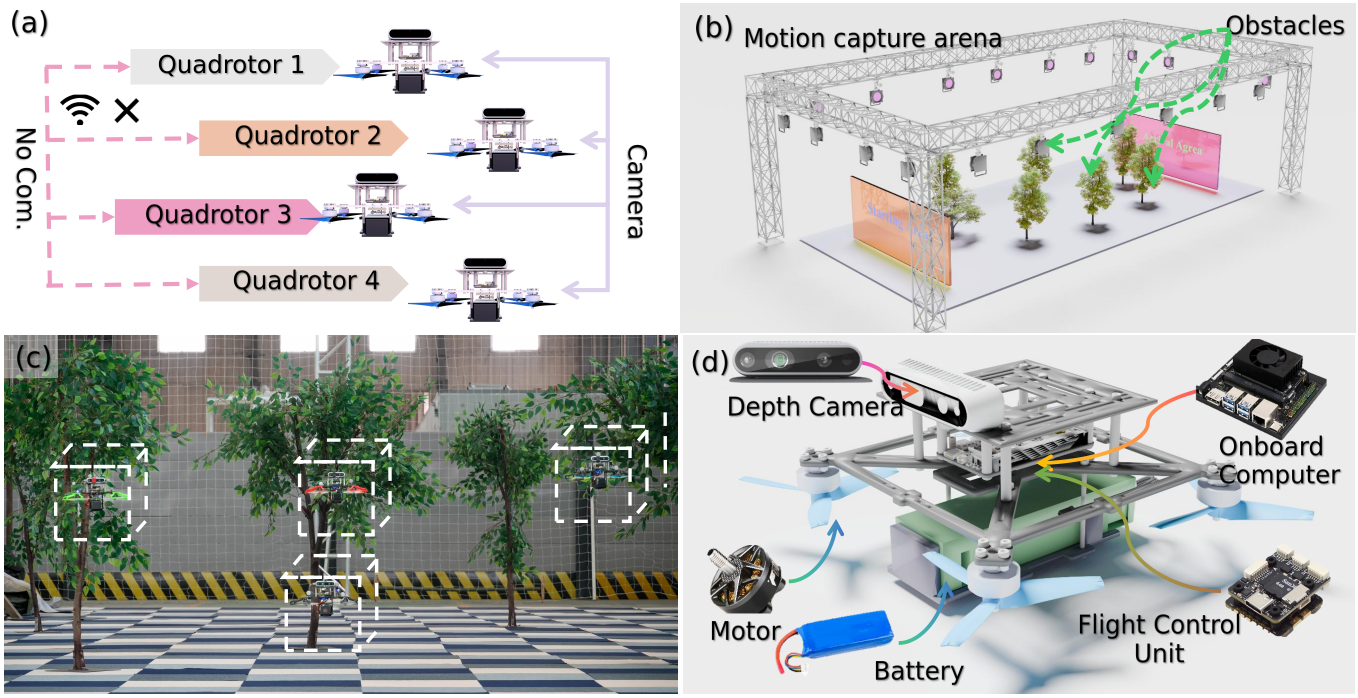

Fig. 4. **Experimental set-up of quadrotors swarm flight in cluttered environments.** a)-(b) Illustrations of the experimental setup and environment configuration. There is no communication connection between quadrotors. They rely solely on onboard sensors and computing units to perform localization, planning, and control. The swarm flies in the three-dimensional (3D) space of an indoor flying arena. The quadrotors take off from randomly assigned initial positions within a predefined start area (orange zone) and fly toward the arrival area (pink zone) on the opposite side of the arena. Note that the motion capture system is used solely to visualize the status of the quadrotors. (c) Indoor test environment with obstacles. (d) Components of the quadrotors employed in the hardware experiments.

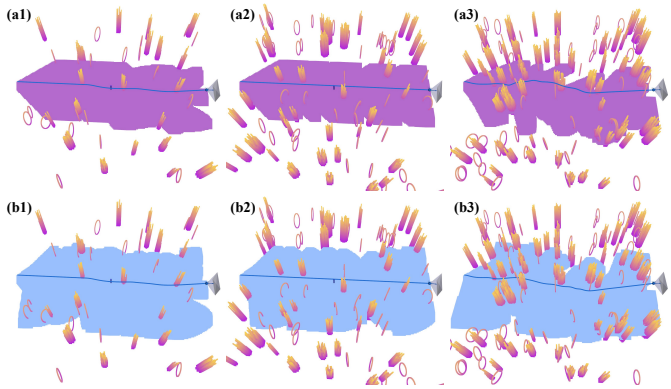

Fig. 5. Environmental observation coverage under varying obstacle densities. The blue line represents the navigation trajectory, while the colored regions indicate the observable space S_{obs} (blue for our method, purple for Zhou's method). (a1)–(a3): Zhou's method perceives the environment only when obstacles appear, leading to limited S_{obs} (purple regions). (b1)–(b3): Our method actively perceives the environment irrespective of obstacle presence, resulting in a substantially larger S_{obs} (blue regions).

2) Evaluation of Target Matching and Tracking: We evaluate the proposed adaptive Kalman filter (AKF) for target matching and tracking through comparative experiments in simulated environments with varying target speeds, using the root mean square error (RMSE) of position and velocity as quantitative metrics (Table **??**).

Compared with the conventional Kalman filter (KF), the proposed AKF consistently achieves lower RMSE in both position and velocity estimation across all evaluated target speeds. At low speed, AKF reduces position RMSE by 57.1% ($0.14\ \text{m} \rightarrow 0.06\ \text{m}$) and velocity RMSE by 13.8% ($0.29\ \text{m/s} \rightarrow 0.25\ \text{m/s}$). More notably, at high speed, AKF yields an 83.6% reduction in position RMSE ($1.40\ \text{m} \rightarrow 0.23\ \text{m}$) and an 88.2% reduction in velocity RMSE ($2.87\ \text{m/s} \rightarrow 0.34\ \text{m/s}$). These results highlight the effectiveness of the adaptive mechanism in handling dynamic targets with varying velocities.

3) Evaluation of Overall MP-CDE System: We compare the proposed method against several state-of-the-art approaches, including EGO-Swarm [5], R-MADER [14] and RBL-Controller [18]. A team of six quadrotors is deployed to navigate within a complex simulated environment, as illustrated in Fig. 7. To comprehensively evaluate performance, the quadrotors are tested across three target velocity regimes: low speed ($0.5\ \text{m/s}$), medium speed ($1.0\ \text{m/s}$), and high speed ($2.0\ \text{m/s}$). As these baseline methods do not explicitly consider communication-denied scenarios, the following adjustments are applied to enable a fair comparison: communication is permitted only when other quadrotors are within the FoV; for R-MADER, a maximum communication delay of 0.5 s is enforced; for RBL-Controller, when no obstacles exist between the quadrotors and their mutual distance is below

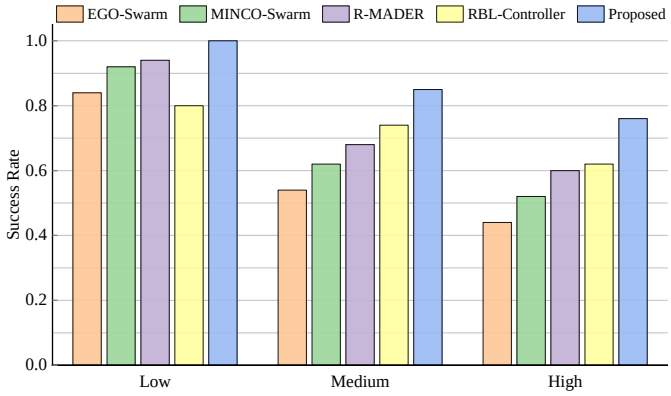

Fig. 6. Comparison of success rates among different methods under varying velocity settings.

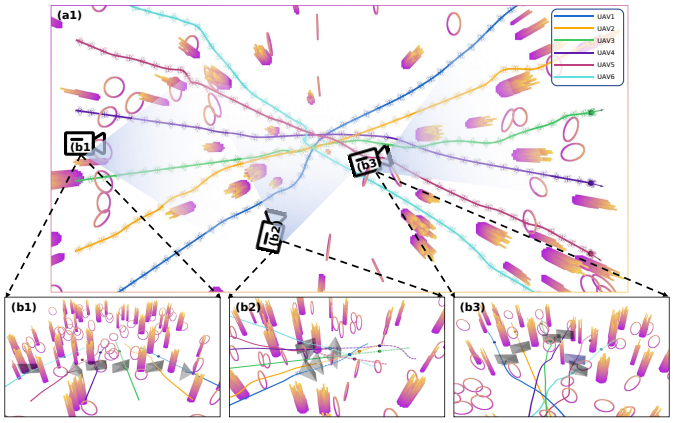

Fig. 7. Swarm system with six quadrotors flying in a dense environment. Curves of different colors represent the historical trajectory of quadrotors.

a predefined threshold, the positions of neighboring quadrotors can be obtained. We additionally re-implemented the communication-free Learning Interaction Aware planner [8] under the same perception stack; however, it repeatedly failed to converge in dense clutter (success rate below 20% across all velocity regimes) because its policy cannot recover from extended occlusions. These failure cases are included in the supplementary material for completeness. All baselines are executed with the identical parameter set reported in Table II so that differences stem solely from the planning logic.

The success rates of all methods under three velocity settings are summarized in Fig. 6. Overall, the proposed method consistently outperforms the baseline approaches across all velocity levels, achieving the highest success rates and demonstrating robustness under communication-denied conditions.

While all methods perform well at low velocities, increasing speed reduces the time available for motion planning, resulting in a decline in success rates across all approaches. At higher velocities, the proposed method maintains superior success rates, highlighting its effectiveness under challenging conditions.

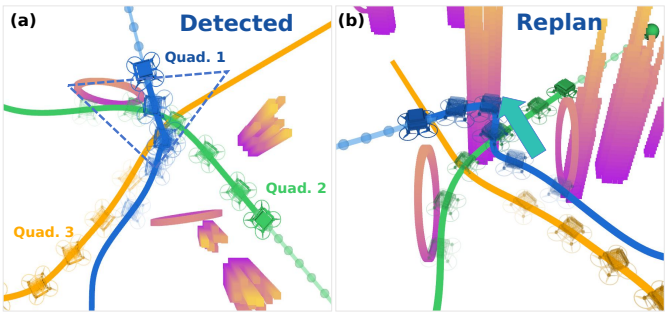

Fig. 8. Replanning details of the swarm system with six quadrotors flying in a dense environment. Curves of different colors represent the historical trajectory of quadrotors.

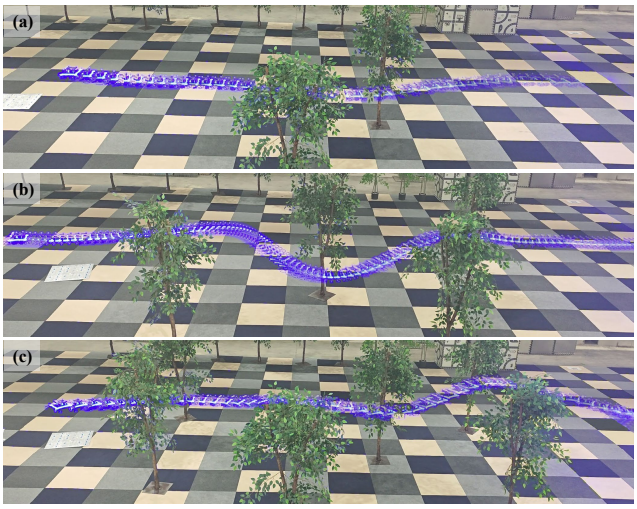

Fig. 9. Active perception of quadrotor under different obstacle densities in real environments. Colored afterimages depict the executed trajectories. The quadrotor increases the range of environmental perception while ensuring safety.

C. Real World Experiment

To validate the proposed method, real-world experiments were conducted for single-robot, ablation, and swarm studies, using VINS-Fusion for positioning and YOLOv11 [30] for quadrotor detection. The FZMotion motion capture system was employed to provide ground truth data for comparative analysis. Throughout all experiments, the quadrotors were operated with a maximum linear velocity of $1\ \mathrm{m/s}$ and a maximum angular velocity of $\pi/2\ \mathrm{rad/s}$, and were equipped with cameras featuring a field of view (FoV) of $85^\circ \times 58^\circ$.

1) Single-robot Experiments: We conducted three single robot autonomous navigation experiments using quadrotors in environments with 2, 4, and 6 obstacles. The goal is set 10 m ahead of the starting position. The quadrotor rapidly avoids obstacles sequentially and simultaneously explores the environment by actively planning its yaw angle while maintaining safe operation. Fig. 5 (a2)–(c2) displays screenshots capturing the entire flight process, while (a1)–(c1) of the same figure show screenshots of the detection area during the flight. These experiments demonstrate the effectiveness and practicality of

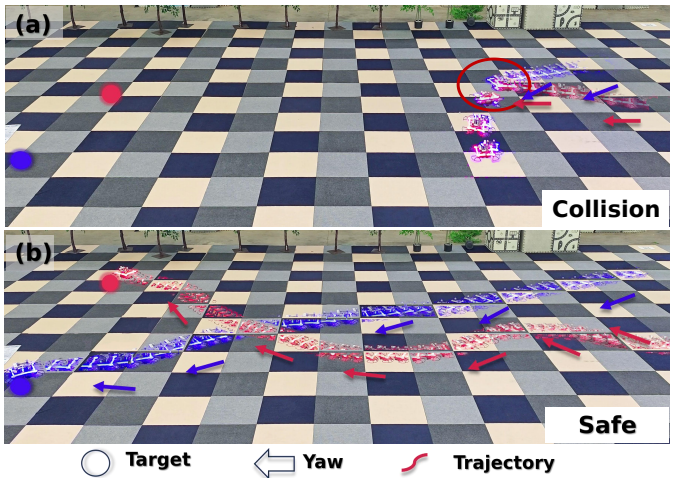

Fig. 10. Ablation study in real-world experiment. Colored traces indicate the executed trajectories. Arrows indicate the heading of each quadrotor. Collisions are highlighted with red ellipses.

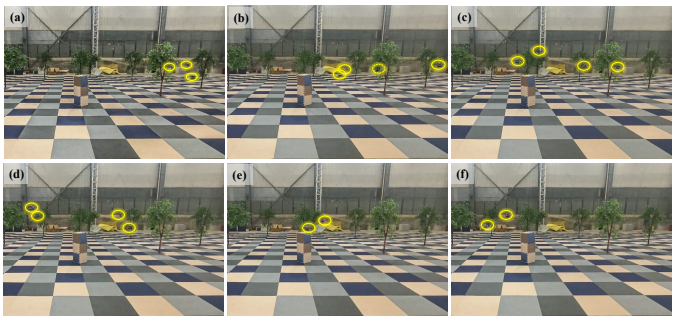

Fig. 11. Snapshots of the swarm flight process with dynamic obstacle avoidance. The letters (a)–(f) in the upper-left corner of each subfigure mark the frame sequence as time increases.

the proposed method for single robot autonomous navigation.

These flights isolate the active yaw module from inter-agent interactions and therefore quantify how much environmental coverage can be achieved by a single platform under the same FoV and velocity constraints used in the swarm studies. The resulting coverage statistics (S_{extobs} and visibility duration) feed directly into the safety-metric calibration employed when multiple vehicles are present.

2) Ablation Study: To evaluate the effectiveness of the proposed method under communication-denied conditions, we conducted a cross-flight experiment, as depicted in Fig. 10. Two quadrotors flew towards goals 8 m apart, crossing each other near the center of the scenario. When the quadrotor moves along the planned trajectory, due to the FoV limitation of the camera, the quadrotors entered each other's blind spots and eventually collided, as shown in Fig. 10 (a). When the yaw angle planning module was enabled, the quadrotors actively explored the environment, detected each other earlier, and avoided collisions, as shown in Fig. 10 (b). The experiment confirms the practicality of the proposed planner in swarm scenarios.

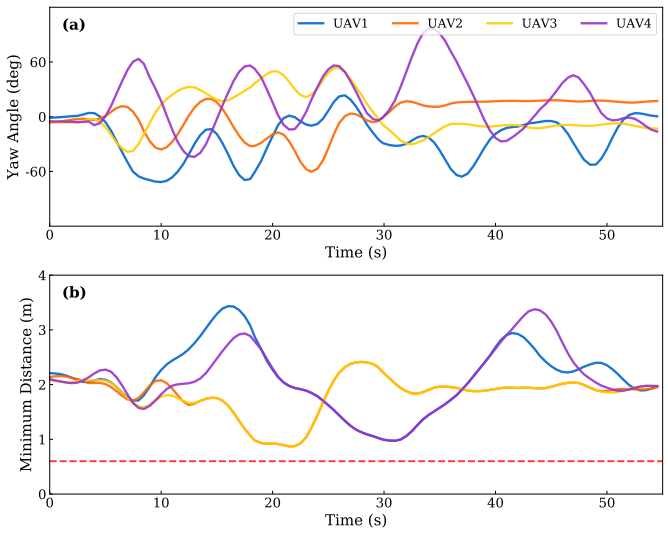

Fig. 12. Minimum inter-quadrotor distance among the four quadrotors and their yaw orientation throughout the experiment. Throughout the experiment, the quadrotors maintain a sufficient safety distance and actively adjust their yaw angles to maintain awareness of neighboring quadrotors. Top: The yaw angle of each quadrotor. Bottom: Minimum distance of each quadrotor from its closest quadrotor.

3) Swarm Experiment: In the real-world swarm navigation experiments, four quadrotors repeatedly executed cross-flight tasks under three obstacle layouts, as illustrated in Fig. 7(b). Each sortie started from origin-symmetric initial poses and converged toward mirrored goal regions while negotiating clutter without any inter-robot communication. The vehicles maintained an average cruise speed of $1\ \mathrm{m/s}$ and relied solely on the onboard perception stack described earlier. For every run we log the full $SE(3)$ trajectories, detection timestamps, and yaw commands; the reconstructed minimum distances and yaw evolution are summarised in Fig. 12. The complete trajectory set is released with the supplementary material, allowing independent verification of the planner's behaviour and quantitative benchmarking against future approaches.

VI. Conclusion

In this paper, we propose a fully autonomous quadrotor swarm motion planning method that enables safe and efficient motion planning in communication-denied environments. By actively and reasonably selecting the yaw angle, the method significantly increases the likelihood of detecting neighboring quadrotors while ensuring safety. Using the dynamically updated Extended Kalman filter and Hungarian algorithm, lightweight trajectory prediction is achieved. Simulation and hardware testing demonstrate the adaptability and effectiveness of MP-CDE in communication-denied environments.

References

[1] X. Liang, H. Yu, Z. Zhang, H. Liu, Y. Fang, and J. Han, "Unmanned aerial transportation system with flexible connection between the quadrotor and the payload: Modeling, controller design, and experimental validation," *IEEE Trans. Ind. Electron.*, vol. 70, no. 2, pp. 1870–1882, 2023.

[2] H. Wang, S. Zhang, X. Zhang, X. Zhang, and J. Liu, "Near-optimal 3-d visual coverage for quadrotor unmanned aerial vehicles under photogrammetric constraints," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1694–1704, 2022.
[3] S. Wang, Y. Wang, Z. Miao, X. Wang, and W. He, "Dual model predictive control of multiple quadrotors with formation maintenance and collision avoidance," *IEEE Trans. Ind. Electron.*, vol. 71, no. 12, pp. 16 037–16 046, 2024.
[4] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi–robot systems," *Curr. Robot. Rep.*, vol. 3, no. 4, pp. 213–225, Aug. 2022.
[5] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, and F. Gao, "Swarm of micro flying robots in the wild," *Sci. Robot.*, vol. 7, no. 66, p. eabm5954, 2022.
[6] J. Hou, X. Zhou, Z. Gan, and F. Gao, "Enhanced decentralized autonomous aerial robot teams with group planning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9240–9247, 2022.
[7] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non–communicating multiagent collision avoidance with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 285–292, 2017.
[8] H. Zhu, F. M. Claramunt, B. Brito, and J. Alonso-Mora, "Learning interaction–aware trajectory predictions for decentralized multi–robot motion planning in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2256–2263, 2021.
[9] H. Zhu, F. M. Claramunt, B. Brito, and J. Alonso-Mora, "Learning interaction–aware trajectory predictions for decentralized multi–robot motion planning in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2256–2263, Apr. 2021.
[10] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on–line collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1047–1054, Apr. 2017.
[11] Q. Lin, Z. Miao, X. Wang, W. He, Y. Wang, and P. Shi, "Collision-free coordination for quadrotors based on differential flatness," *IEEE Trans. Ind. Electron.*, pp. 1–12, 2025.
[12] J. Tordesillas and J. P. How, "Mader: Trajectory planner in multiagent and dynamic environments," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 463–476, 2022.
[13] J. Hou, X. Zhou, N. Pan, A. Li, Y. Guan, C. Xu, Z. Gan, and F. Gao, "Primitive–swarm: An ultra–lightweight and scalable planner for large–scale aerial swarms," 2025. [Online]. Available: https://arxiv.org/abs/2502.16887
[14] K. Kondo, R. Figueroa, J. Rached, J. Tordesillas, P. C. Lusk, and J. P. How, "Robust mader: Decentralized multiagent trajectory planner robust to communication delay in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1476–1483, 2024.
[15] C. Toumieh and D. Floreano, "High–speed motion planning for aerial swarms in unknown and cluttered environments," *IEEE Trans. Robot.*, vol. 40, no. 12, pp. 3642–3656, 2024.
[16] B. Senbaslar and G. S. Sukhatme, "Dream: Decentralized real–time asynchronous probabilistic trajectory planning for collision–free multi-robot navigation in cluttered environments," *IEEE Trans. Robot.*, vol. 41, pp. 573–592, Jan. 2025.
[17] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi–robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, 2020.
[18] L. Pan, M. Catellani, L. Sabattini, and N. Ayanian, "Robust trajectory generation and control for quadrotor motion planning with field–of–view control barrier certification," *arXiv preprint arXiv:2502.01009*, 2025.
[19] M. Boldrer, V. Kratky, V. Walter, and M. Saska, "Swarming in the wild: A distributed communication–less lloyd–based algorithm dealing with uncertainties," *arXiv preprint arXiv:2504.18840*, 2025.
[20] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego–planner: An esdf–free gradient–based local planner for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 478–485, Apr. 2021.
[21] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 922–938, Apr. 2022.
[22] B. Senbaslar and G. S. Sukhatme, "Asynchronous real–time decentralized multi–robot trajectory planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2022.
[23] X. Chen, Y. Zhang, B. Zhou, and S. Shen, "Apace: Agile and perception–aware trajectory generation for quadrotor flights," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 17 858–17 864, May. 2024.
[24] C. Yu, Z. Lu, J. Mei, and B. Zhou, "Perception–aware planning for quadrotor flight in unknown and feature–limited environments," 2025.
[25] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May. 2011.
[26] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception–aware trajectory replanning for quadrotor fast flight," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1992–2009, Dec. 2021.
[27] M. Lu, X. Fan, H. Chen, and P. Lu, "Fapp: Fast and adaptive perception and planning for uavs in dynamic cluttered environments," *IEEE Trans. Robot.*, vol. 41, pp. 871–886, Jan. 2025.
[28] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3259–3278, Oct. 2022.
[29] J. Zeng, Y. Wang, Z. Miao, W. He, and H. Wang, "Decentralized trajectory planning for formation flight in unknown and dense environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 8990–8997, Oct. 2024.
[30] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," 2024. [Online]. Available: https://arxiv.org/abs/2410.17725

Xingxun Liu received the B.E. degree in automation from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2024. He is currently working toward the Ph.D. degree in the School of Artificial Intelligence and Robotics, Hunan University.

His current research interests include motion planning, reinforcement learning, and cluster planning.

Yaonan Wang received the Ph.D. degree in electrical engineering from Hunan University, Changsha, China, in 1994. He is currently an Academician with the Chinese Academy of Engineering, Beijing, China. From 1994 to 1995, he was a Postdoctoral Research Fellow with the National University of Defence Technology, Changsha, China. From 1998 to 2000, he was a Senior Humboldt Fellow in Germany. From 2001 to 2004, he was a Visiting Professor with the University of Bremen, Bremen, Germany. Between 2001 and 2020, he was the Dean of the College of Electrical and Information Engineering, Hunan University. His research interests include robotics, intelligent control, and machine vision.

Zhiqiang Miao received the B.S. and Ph.D. degrees in electrical and information engineering from Hunan University, Changsha, China, in 2010 and 2016. He is currently an Professor with the School of Artificial Intelligence and Robotics, Hunan University. From 2014 to 2015, he was a Visiting Scholar with the University of New Mexico, Albuquerque, NM, USA. From 2016 to 2018, he was a Postdoctoral Fellow with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include multirobot systems, visual navigation, and nonlinear control.

Wei He (Fellow, IEEE) received his B.Eng. and his M.Eng. degrees from College of Automation Science and Engineering, South China University of Technology (SCUT), China, in 2006 and 2008, respectively, and his PhD degree from Department of Electrical Computer Engineering, the National University of Singapore (NUS), Singapore, in 2011. He is currently working as a professor in School of Automation and Institute of Artificial Intelligence, Beijing Information Science and Technology University, and School of Intelligence Science and Technology and Institute of Artificial Intelligence, University of Science and Technology Beijing, China.

He is a Fellow of IEEE. He has co-authored 4 books and published over 300 international journal and conference papers. He is a highly cited researcher by Clarivate Analytics from 2019 to 2024. He is serving as the Senior Editor of IEEE Transactions on Systems, Man, and Cybernetics: Systems. He was served/is serving as Associate Editor of IEEE Transactions on Robotics, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Control Systems Technology, IEEE Transactions on Automation Science and Engineering, SCIENCE CHINA Information Sciences, IEEE/CAA Journal of Automatica Sinica. His current research interests include robotics and intelligent control systems.

Xiangke Wang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2004, 2006, and 2012, respectively. Since 2012, he has been a Lecturer, an Associate Professor, and a Professor with the College of Intelligence Science and Technology, NUDT. From 2009 to 2011, he was a Visiting Student with Australian National University, Canberra, ACT, Australia, supported by the China Scholarship Council. He has authored or co-authored two books and more than 100 publications in peer-reviewed journals and international conferences His research interests include multiagent systems, nonlinear control, and UAVs.