IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.
Cite as T-ASE paper.

Nav-SCOPE: Swarm Robot Cooperative Perception and Coordinated Navigation

Chenxi Li, *Graduate Student Member, IEEE*, Weining Lu, Qingquan Lin, Litong Meng, *Member, IEEE*, Haolu Li, and Bin Liang, *Senior Member, IEEE*

***Abstract*—This paper proposes a lightweight decentralized solution for multi-robot coordinated navigation with cooperative perception. First, we introduce a rapid way to process sensory data, thus obtaining safe directions and key environmental features. Then, an information flow is created to facilitate real-time perception sharing over wireless ad-hoc networks. Consequently, the environmental uncertainties of each robot are reduced by interaction fields that deliver complementary information. Finally, path optimization is achieved in a probabilistic way, enabling self-organized coordination with effective convergence, divergence, and collision avoidance. Our method is fully interpretable and ready for deployment without gaps. Comprehensive simulations and real-world experiments demonstrate reduced path redundancy, robust performance across various tasks, and minimal demands on computation and communication.**

***Note to Practitioners*—Local perception is the information source for robot navigation in unknown environments. The extended perception at the swarm level provides complementary information for each robot. This can optimize robot paths and achieve coordinated navigation in a distributed way. However, this purpose is currently unachievable due to the high demands of communication and computation on board. This study reduces these two demands to the extreme by Fast Fourier Transform (FFT), digital filtering, probabilistic fusion, and swarm interaction forces. At the same time, our fully interpretable method avoids sim-to-real gaps and can be deployed directly on ground mobile robots. Simulation and real-world experiments outperform a state-of-the-art approach. Future work aims to extend our method to aerial robots as drones.**

***Index Terms*—Multiple mobile robots, cooperative systems, distributed robot systems.**

I. Introduction

In RECENT years, multi-robot systems have shown great potential in wild [1], industry [2], and indoor services [3]. Self-organized swarms stand out for their better automation and wider application. Despite significant progress, most solutions only use each robot's local perception independently for navigation, while few studies have so far utilized real-time cooperative perception of the swarm to optimize the planning process. Consequently, robots' coordination is often confined to leader-follower [4], [5] or collision avoidance [1], [6] due to limited information flow within swarm systems, in which robots often encounter unexpected obstructions or unnecessary detours.

The advantages of cooperative perception are obvious: a wider field of view, more details of the unknown environment, and that lead to better paths with fusion optimization. The challenges include but are not limited to informative environmental representation, decision-making with fused perception, on-board processing latency, and minimal unreliable communication. The first two are method requirements, and the latter two are hardware constraints.

This paper proposes Nav-SCOPE: **S**warm **CO**operative **PE**rception and coordinated **Nav**igation. This is an extremely lightweight systematic solution that addresses the above contradiction. In this system, each robot is used as a telescope of neighboring robots for an extended field of view. When robots decide their future paths, they will use the fused perception to choose better directions with fewer detours or conflicts. Experimental results show self-organized paths and better swarm automation. To the best of our knowledge, Nav-SCOPE is the first interpretable navigation solution with real-time cooperative perception, and it does not require a mapping process or external localization. The paper makes the following contributions:

- The formulation of perception and fusion is established systematically and interpretively with minimal computation and communication requirements.
- The navigation process is detailed for better swarm coordination and less path redundancy. Parameters can be determined directly before deployment.
- Comprehensive simulations in challenging environments demonstrate that Nav-SCOPE outperforms a previously state-of-the-art approach. Real-world experiments show that our method can be applied robustly without gaps.

This article is organized as follows. Section II presents the related work. Section III gives the preliminaries of our work. In Section IV, the formulation of Nav-SCOPE is detailed, including sensor data processing, information flow from cooperative perception, coordinated navigation, and path

Received 13 May 2025; revised 24 July 2025; accepted 26 August 2025. Date of publication 29 August 2025; date of current version 9 September 2025. This article was recommended for publication by Associate Editor Y. Li and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported in part by the Beijing National Research Center for Information Science and Technology (BNRist) Project under Grant BNR2024TD03003 and in part by the National Natural Science Foundation of China under Grant 92248304. *(Corresponding author: Weining Lu.)*

Chenxi Li, Qingquan Lin, Litong Meng, and Bin Liang are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: lcx22@mails.tsinghua.edu.cn; linqq19@tsinghua.org.cn; menglt@ieee.org; liangbin@tsinghua.edu.cn).

Weining Lu is with Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: luwn@tsinghua.edu.cn).

Haolu Li is with the School of Mechanical and Electrical Engineering, Beijing Information Science and Technology University, Beijing 100192, China (e-mail: 202402005@bistu.edu.cn).

Digital Object Identifier 10.1109/TASE.2025.3604178

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

optimization. The necessary steps for the deployment of Nav-SCOPE are introduced in Section V. Simulation and real word experiments are presented in Section VI and Section VII, respectively. Finally, Section VIII presents the conclusions of this study.

II. Related Works

Numerous studies have focused on local planning since the advent of mobile robots. Autonomous navigation was first realized in individual robots. Reactive strategies, such as Potential Fields (PFs) [7], [8] and Bugs [9], [10], are direct and lightweight solutions. Planners based on voxel maps [11], [12] can provide better local trajectories with more environmental details. Learning-based methods such as reinforcement learning [13] and guidance planning [14] have also excelled for their automation and become popular in recent years. These methods do not require a pre-built map, as a Simultaneous Localization and Mapping (SLAM) [15] process before planning execution is not suitable for many tasks and cases.

As robot systems grow to fleets and swarms, critical issues of coordination between robots need to be solved. Centralized control is a typical way to regulate robot paths. For example, [8] and [16] used central computing stations to achieve flock flight on multiple quadrotors. The former is a typical solution based on optimization, while the latter is achieved by reinforcement learning. However, these approaches require ideal communication and external localization, making them vulnerable in scaled-up scenarios [17]. References [2] and [3] used received signal strength indication (RSSI) to achieve distributed navigation, but still needed external beacons to provide a priori environmental information, which limits the scalability of the algorithm.

Decentralized systems can be more versatile in large unknown environments. The more information utilized, the better performance a system would have. Currently, it is common to use mutual positioning and trajectory sharing to achieve self-organization and swarm navigation. Consequently, recent studies have provided two types of solutions. (i) Leader-follower structures achieve synergic formation towards the same goal, such as leading robot following [18], formation control [4], [5] and target tracking [19], [20], which are mainly based on distance keeping and barycentric coordinates. (ii) Collision avoidance structures optimize robot paths to reduce negative interactions in various tasks, such as spatial-temporal optimization of trajectories [1], [21], and similar approaches performed by neural networks [6], [22]. A typical state-of-the-art (SOTA) solution is EGO-Swarm-v2 [1], which was tested in wild bamboo forests.

Cooperative perception offers external environmental information to each robot, thereby holding significant potential to enhance swarm navigation and improve their performance. However, it is challenging to form an information flow among robots, as well as to take advantage of it. References [23] and [24] proposed Graph Neural Network (GNN) pipelines to share perceptual features, while the swarm planning process was not yet considered. Reference [25] achieved a systematic GNN framework from perception to planning execution, but robot actions were limited to discrete on grid maps, thereby suffering from sim-to-real gaps. In contrast, Nav-SCOPE is designed to solve these challenges and can be immediately deployed in real-world applications.

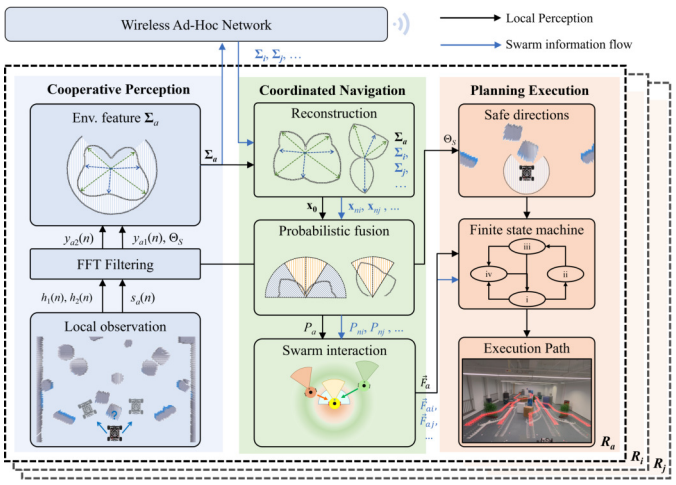

Fig. 1. System architecture.

III. Preliminaries

We consider a swarm of ground mobile robots $R_i, i \in \{1, 2, \ldots, N\}$ operating in unknown cluttered environments. Each robot has its local perception within the maximum range of R. The sensory data is defined as a digital depth sequence $s_i(n)$, with M uniform samples within its field of view (FOV):

$$s_i(n) = \mathbf{d}\left[\theta_n \in \Theta\right] \leq R, \quad n = 1, 2, \ldots, M, \tag{1}$$

where Θ is the FOV, and $\mathbf{d}$ is the sequent depth array obtained from LiDARs or cameras, with each item corresponding to a distance in a certain direction θ_n. Consider the physical dimension r_0 of a robot, a safety margin r is used to keep the robot from obstacles, as shown in Fig. 2. The FOV is divided into four quadrants: navigation scope (Q_1, Q_2), and peripheral vision (Q_3, Q_4). They are used to evaluate the occupation of the future route. The scope angle α and planning distance l_{th} are determined by geometric relationships:

$$\alpha = \pi - 2\arccos\left(\frac{r_0}{r}\right), \quad l_{th} = \frac{r}{\cos\left(\frac{\pi-\alpha}{2}\right)} = \frac{r^2}{r_0}. \tag{2}$$

The necessary and sufficient condition of a safe direction can be expressed as a safety window in Q_1 and Q_2:

$$\theta_m \text{ is safe} \iff \text{for } n \in \left(m - \frac{\alpha}{2}, m + \frac{\alpha}{2}\right), \; s_i(n) \geq l_{th}. \tag{3}$$

In this study, each robot can advance or rotate left and right towards a safe direction. The task of the swarm is to navigate safely and coordinately towards targets. The objective of Nav-SCOPE is to provide forward directions to robots, thus minimizing swarm path lengths with the least collision and resource consumption.

IV. Formulation of Nav-SCOPE

The system architecture of Nav-SCOPE is illustrated in Fig. 1. In this section, we will first introduce the FFT process

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

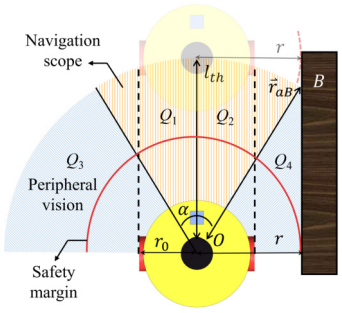

Fig. 2. Safe margin and four observation quadrants. If there is no obstacle within the orange scope, the robot can keep a safety margin r from the obstacle B.

and our filter design. Then, the information flow among swarm robots is formed. Finally, coordinated navigation facilitated by cooperative perception is elaborated.

A. FFT Process and Filter Design

In the perception stage, Nav-SCOPE scans to evaluate all directions of $s_i(n)$ within the FOV. The evaluation process can be performed rapidly by digital filtering based on Fast Fourier Transform (FFT), which was demonstrated in a previous study [26], since convolution in the time domain is equivariant to multiplication in the frequency domain. The overall complexity is $O(m \log m)$ in computation, and an illustration of the framework is shown in Fig. 3 (i)-(iv). The evaluation of this stage serves two functions: first, to identify safe routes and second, to extract key features of the surrounding environment.

The design of digital filters is critical for FFT filtering. In order to achieve the two functions discussed above, we propose two types of filters, respectively. First, to identify safe routes subject to robot dimensions r_0 and planning distance l_{th}, a normalized rectangular window can be designed to evaluate the occupancy of each direction, which is mathematically equivalent to (3). The size of the window T_c corresponds to α in the physical world:

$$h_1(n) = \frac{1}{T_c} R_{T_c}(n-\tau), \quad n = 1, 2, \ldots, M, \tag{4}$$

where $T_c = M(\alpha/\Theta)$ refers to the window size normalized by the digital resolution, and $\tau = \frac{M-1}{2}$ is the group delay. This filter operates on short-term perception within the planning distance, so $s_i(n)$ is truncated to l_{th} before being filtered. The safety condition (3) is further simplified with the filtered sequence $y_{i1}(n)$:

$$\theta_m \text{ is safe} \iff y_{i1}(m) \geq l_{th}, \tag{5}$$

and we obtain the set of safe directions $\Theta_s = \{\theta_m | y_{i1}(m) \geq l_{th}\}$ for local planning, e.g., the sector in Fig. 3 (iii).

Second, Nav-SCOPE is required to evaluate environmental occupancy at the level of perceptual range in the long term. A low-pass filter can effectively extract overall spacial tendency, while filtering out minor details for subsequent compression. Consequently, another filter h_2 is designed to extract the key features of open space and potential obstructions according to the same window size T_c:

$$h_2(n) = 2 f_c \operatorname{sinc}\left(2 f_c (n-\tau)\right) w_M(n, \beta), \tag{6}$$

where $f_c = 1/T_c$, $\operatorname{sinc}(x) = sin(\pi x)/(\pi x)$, and $w_M(n, \beta)$ is a Kaiser window [27] with parameter β. As illustrated in Fig. 3(iv), the filtered sequence $y_{i2}(n)$ contains several sectors. The centers of these sectors are prospective pathways in long-term planning, and the areas are approximate estimates of open spaces.

B. Information Flow From Cooperative Perception

The information flow within the swarm is formed by distributed sharing of robot perception. In order to minimize communication loads, Nav-SCOPE performs informative encoding at transmitter, and rapid reconstruction at receiver. The centers and borders of the sectors in $y_{i2}(n)$ indicate possible open spaces and potential obstacles, respectively. Therefore, they are key perceptual features encoded for data transmission. The features can be obtained by calculating the local extrema of $y_{i2}(n)$, which are then masked by safe directions Θ_s to filter out narrow spaces relative to robot size (Fig. 3 (vi)-(vii)):

$$\mathbf{\Sigma}_i = \left\{(\phi_{i,j}, d_{i,j}) | \Delta y_{i2}(j) \Delta y_{i2}(j+1) < 0 \cap \Theta_s\right\}, \tag{7}$$

where $\phi_{i,j}$ and $d_{i,j}$ are the directions and distances of the extrema. Consequently, we get perceptual features of robots only at the byte level. At the receiver, the directions $\phi_{i,j}$ should be calibrated according to the relative pose of two robots. Then, we use simple linear interpolation for approximate reconstruction, as shown in Fig. 3 (viii), and the information flow within the swarm is established.

In our applications in unknown, cluttered environments, robots are often obstructed by obstacles. The objective of the information flow is to deliver complementary observation of the surrounding environment. Consequently, an effective expansion should keep an overlap between the perceptual regions of two robots. This condition ensures correlation and accessibility between these observations. Hence, robots only connect to the neighboring robots within the FOV, which does not need a strongly connected network, but is time-varying and forms local groups spontaneously. In addition, the information flow of Nav-SCOPE is event-triggered and in an ad hoc way. When a robot needs cooperative perceptions from its neighbors, it will request data transmissions, and then the information flows in a one-way direction to the robot in need. In summary, the transmission mechanism is as follows:

- First, cooperative perception activates only among neighboring robots within visual range. This can prevent irrelevant information from robots too far away.
- Second, a visual persistence mechanism is introduced to address temporary visual loss towards neighbors.
- Third, the observation shares only when needed, such as the time to decide future navigation directions.

Thus, the data are minimized while sufficient information is maintained, and the complexity of transmission scales linearly

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

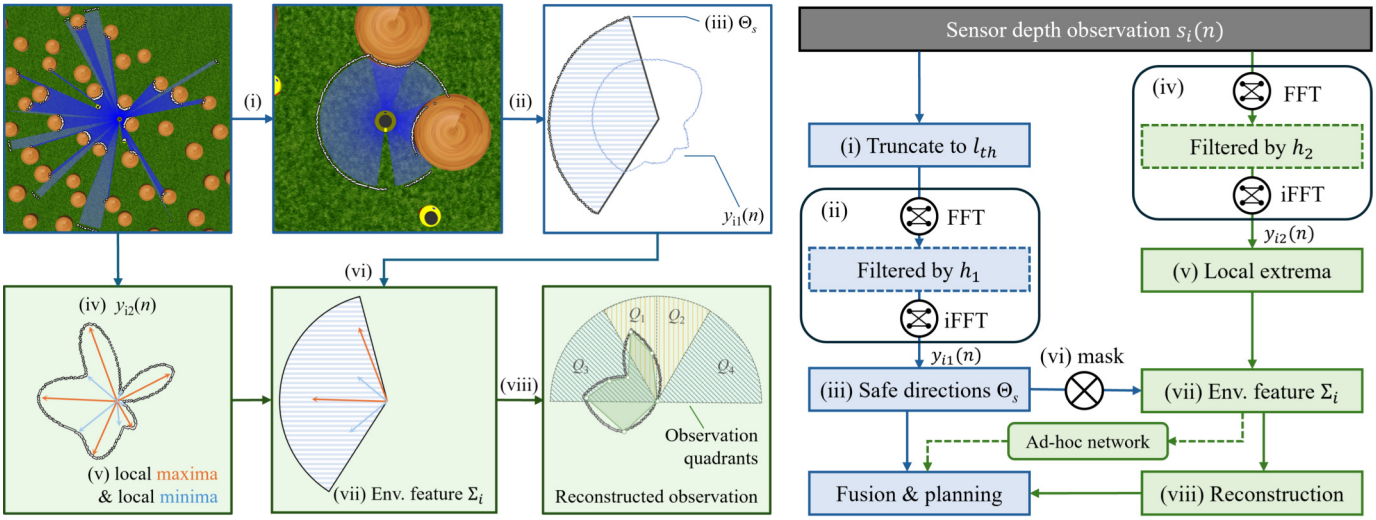

Fig. 3. Sensor data processing framework (right) and an illustrative example (left) in local coordinate. Blue boxes identify safe directions and green boxes extract environmental features. The feature excludes narrow spaces beneath the robot dimension, while maintains prospective directions for reconstruction. The computation complexity in (ii) and (iv) is $O(m \log m)$, while others are less than $O(m)$. The volume of environmental feature Σ_i is usually less than $(2\Theta/\alpha)$.

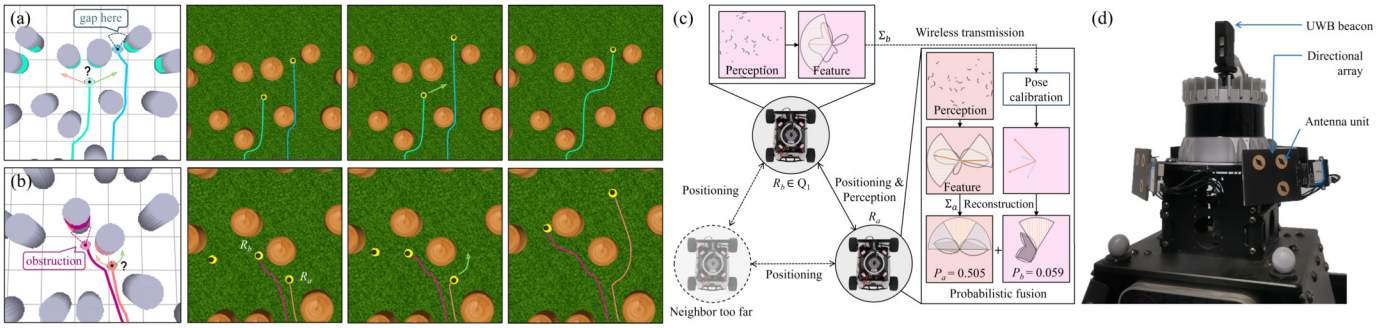

Fig. 4. Detailed cases of coordination in simulation. (a) Convergence case. (b) Divergence case. (c) A detailed illustration of fusion optimization in (b). R_a has similar probabilities in either directions before fusion. The perception of it's neighbor R_b in quadrant Q_1 provides strong probability of obstruction. Consequently, R_a choose to navigate rightwards. (d) Distributed ultra-wide band (UWB) model of each robot for communication and mutual positioning.

with the number of observed neighbors. The reduction in communication resource improves reliability under rigorous conditions or unreliable networks.

C. Coordinated Navigation and Path Optimization

The objective of coordinated navigation is to optimize robot paths in a distributed way, combining local perception with swarm information flow. Here, neighboring robots perform as telescopes to provide complementary information in terms of the environment, in other words, help to find open spaces with fewer obstacles towards the target. We use forces to indicate the level of impact.

1) Individual Intelligence: Robots use their local perception to evoke a tendency in a certain direction within Θ_s. Considering a robot R_a, we use a force $\vec{F}_a$ to quantify its motivation. Note that there are two types of turning directions, namely left and right, for ground robots, the amplitude of $\vec{F}_a$ is designed as the probability of searching its path on the left side. If there is no obstruction within the planning distance l_{th}, R_a will head towards the target direction $\vec{e}_a$ with the same probability on both sides. Otherwise, when an obstacle is detected in the navigation scope, R_a should select a direction for circumnavigation:

$$\vec{F}_a = P_a \vec{e}_a, \tag{8}$$

where P_a uses a sigmoid function $\sigma(\cdot)$ to map local perception into the probability domain:

$$P_a = \begin{cases} \sigma[\mathbf{W}_0^\top \mathbf{x}_0 + b_0] & \exists \text{ obstacle} \\ 0.5 & \text{other cases,} \end{cases} \tag{9}$$

with open space areas $\mathbf{x}_0$ from the reconstructed local Σ_a, and the weight vector $\mathbf{W}_0$ in the four quadrants. The bias entry b_0 is usually set to zero without a priori. The values of $\mathbf{W}_0$ assign weights to observations in both the navigation scope and the peripheral vision. In our experiments, they are equally weighted with odd symmetry left and right.

2) Interaction Fields: They are connections among robots that deliver information flow, regulate swarm coordination, and achieve path optimization. Robots observe their neighbors within the FOV in an ad-hoc way, so we define the interaction field only exerted from local observed neighbors. In Nav-SCOPE, robot connectivity is time-varying according to the visibility and distances of neighboring robots. Consider

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

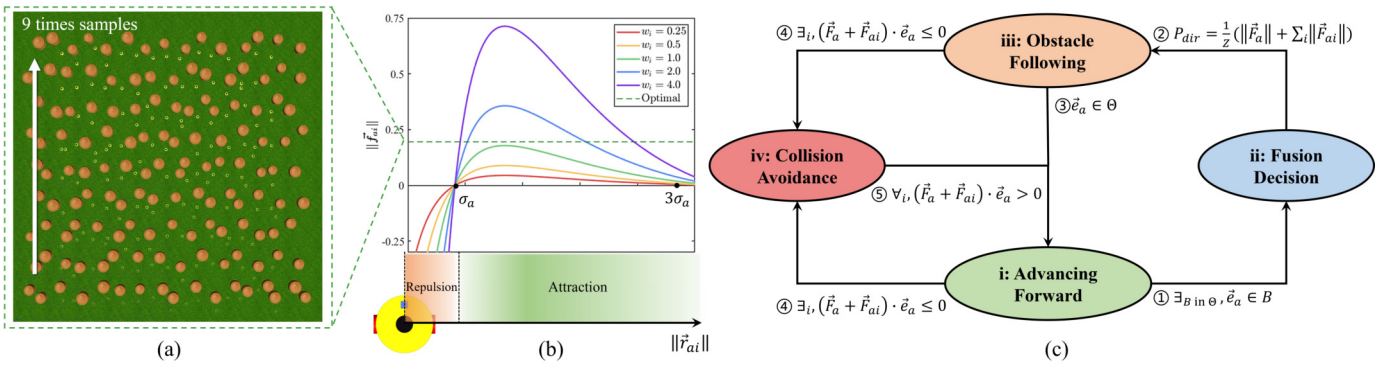

Fig. 5. (a) Sampling process to determine the optimal w_i. (b) Magnitude curves of the interaction fields. (c) Finite state machine and its state transitions.

a robot R_a with its neighbor R_i. Consequently, the influence of neighbor R_i is closely related to its observation features Σ_i and their relative distances $\|\vec{r}_{ai}\|$. The potential V_{ai} of the interaction combines the effect of coordination convergence V_{ai}^{att} and collision avoidance V_{ai}^{rep} [28]:

$$V_{ai}\left(\left\|\vec{r}_{ai}\right\|\right) = V_{ai}^{att}\left(\left\|\vec{r}_{ai}\right\|\right) + V_{ai}^{rep}\left(\left\|\vec{r}_{ai}\right\|\right). \tag{10}$$

To achieve coordination over a moderate distance and implement avoidance when robots are in close proximity, we model the attractive potential with an exponential kernel and the repulsive potential with an inverse proportional kernel:

$$V_{ai}^{att}\left(\left\|\vec{r}_{ai}\right\|\right) = V_0 e^{-\frac{\|\vec{r}_{ai}\|}{\sigma_a}},\ V_{ai}^{rep}\left(\left\|\vec{r}_{ai}\right\|\right) = -\frac{\sigma_a}{\left\|\vec{r}_{ai}\right\|}, \tag{11}$$

where σ_a is the reference distance, marking the transition from repulsion to attraction. The force field $\vec{f}_{ai}$ is the negative gradient of the potential with a weight of w_i. When $\|\vec{r}_{ai}\| = \sigma_a$, $\vec{f}_{ai} = \mathbf{0}$, so the formulation of interaction fields is as follows:

$$\begin{aligned} V_{ai}\left(\left\|\vec{r}_{ai}\right\|\right) &= e^{1-\frac{\|\vec{r}_{ai}\|}{\sigma_a}} - \frac{\sigma_a}{\left\|\vec{r}_{ai}\right\|}, \\ \vec{f}_{ai}\left(\left\|\vec{r}_{ai}\right\|\right) &= -w_i \nabla_{\vec{e}_{ai}} V_{ai} = \frac{w_i}{\sigma_a}\left(e^{1-\frac{\|\vec{r}_{ai}\|}{\sigma_a}} - \frac{\sigma_a^2}{\|\vec{r}_{ai}\|^2}\right)\vec{e}_{ai}, \end{aligned} \tag{12}$$

where $\vec{e}_{ai}$ is a unit vector from R_a to R_i. The amplitude of $\|\vec{f}_{ai}\|$ is depicted in Fig. 5(b). With an increase in w_i, the impact of neighbors becomes more pronounced. The optimal value of w_i can be obtained in Section V-C, and its robustness is demonstrated in simulations. Furthermore, when $0 < \|\vec{r}_{ai}\| \leq \sigma_a$, the interaction force behaves as a repulsion to avoid path interference. When $\sigma_a < \|\vec{r}_{ai}\| \leq 3\sigma_a$, the force is gravitational for cooperative perception. When $\|\vec{r}_{ai}\| > 3\sigma_a$, the force decays almost to 0. We define $\|\vec{r}_{ai}\| \in (\sigma_a, 3\sigma_a]$ as the pass band for observation fusion. Neighboring robots outside the pass band are either too close or too far away. A segmentation function $\vec{F}_{ai}$ is designed to perform differently at varying distances:

$$\vec{F}_{ai} = \begin{cases} \vec{f}_{ai} & \|\vec{r}_{ai}\| \in (0, \sigma_a] \\ P_{ni}\vec{f}_{ai} & \|\vec{r}_{ai}\| \in (\sigma_a, 3\sigma_a] \\ \mathbf{0} & \|\vec{r}_{ai}\| \in (3\sigma_a, +\infty), \end{cases} \tag{13}$$

where P_{ni} is the probability of following the neighbor in future paths. P_{ni} will be high if the neighbor R_i observes few obstacles towards the target of R_a. This will result in a strong attractive effect, thus increasing the likelihood of R_a to follow it. In contrast, if the location of R_i is blocked by obstacles, P_{ni} will be lower, leading to a divergence effect. In such cases, R_a will be more likely to take a different path, which avoids future detours that would otherwise be caused by its observational limitation. The definition of P_{ni} is differentiated by the quadrant in which the neighbor is observed, in accordance with left searching in (9):

$$P_{ni} = \begin{cases} \boldsymbol{\sigma}[\mathbf{1}^\top \cdot (\mathbf{x}_{ni} + \mathbf{w}_n)] & R_i \in \{Q_1, Q_3\} \\ 1 - \boldsymbol{\sigma}[\mathbf{1}^\top \cdot (\mathbf{x}_{ni} + \mathbf{w}_n)] & R_i \in \{Q_2, Q_4\}, \end{cases} \tag{14}$$

where $\mathbf{x}_{ni}$ indicate the estimates of the passable distances towards the target, and $\mathbf{w}_n$ are the distance thresholds of following. They are both 2D vectors corresponding to the scope quadrants (Q_1 and Q_2), after receiving and calibrating the neighboring perception Σ_i.

3) Fusion Optimization: The interaction fields provide additional information to identify more efficient paths with fewer detours. When R_a encounters an obstacle, it combines its own intelligence $\vec{F}_a$ and the resultant of $\vec{F}_{ai}$ for subsequent planning directions:

$$\begin{aligned} P_{dir} &= \frac{1}{Z}\left(\|\vec{F}_a\| + \sum\nolimits_{\vec{F}_{ai}\cdot\vec{e}_{ai}>0} \|\vec{F}_{ai}\|\right), \\ Z &= 1 + \sum\nolimits_{\vec{f}_{ai}\cdot\vec{e}_{ai}>0} (\|\vec{f}_{ai}\|), \end{aligned} \tag{15}$$

where dir = {left, right} for ground robots, P_{dir} refers to the probability of exploration to the left, and Z is a normalization factor. An illustrative example can be found in Fig. 4. The decision is based on the principle of Maximum A Posteriori (MAP) to address the subtle slope of the sigmoid function. The fusion process expands the field of view of individual robots, thus significantly reducing the uncertainty of environmental observations. Consequently, redundant paths are eliminated by convergence or divergence effects in terms of robot paths.

V. Model Deployment

This section details the necessary steps for the deployment of Nav-SCOPE, including the planning process, the mutual positioning of swarm robots, and the determination of parameters.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

Algorithm 1 Nav-SCOPE

1 Initialize neighbor indicators $\mathcal{N}_a, \mathcal{R}_a \leftarrow \varnothing$
2 $\mathcal{S}_a \leftarrow$ CurrentRobotState() in Fig. 5(c) /* Identifying safe directions */
3 FFT points: $N_f \leftarrow 2^{\lceil \log_2 M \rceil}$
4 $\widetilde{s_a}(n) \leftarrow$ Truncate $s_a(n)$ within $[0, l_{th}]$
5 $y_{a1}(m) \leftarrow \mathscr{F}_{N_f}^{-1}\left[\mathscr{F}_{N_f}[\widetilde{s_a}(n)] \odot H_1\right] R_M(m)$
6 $\Theta_s \leftarrow \{\theta_m | y_{a1}(m) \geq l_{th}\}$ /* Updating neighboring robots */
7 **for** R_i in swarm robots **do**
8 **if** Received feature Σ_a request **then**
9 $\mathcal{R}_a$.add(R_i)
10 **end if**
11 **if** R_i is observed within the FOV **then**
12 $\mathcal{N}_a$.add(R_i)
13 Obtain $\vec{r}_{ai}$ and calculate $\vec{f}_{ai}$ from (12)
14 **end if**
15 **end for**/* Event-triggered information flow */
16 **if** $\mathcal{S}_a$ == ii **then**
17 Request and receive features Σ_i from $R_i \in \mathcal{N}_a$
18 Calculate P_{ni} from (14)
19 **else**
20 $P_{ni} = 1$
21 **end if**
22 **if** $\mathcal{S}_a$ == ii or $\mathcal{R}_a \neq \varnothing$ **then**
23 $y_{a2}(m) \leftarrow \mathscr{F}_{N_f}^{-1}\left[\mathscr{F}_{N_f}[s_a(n)] \odot H_2\right] R_M(m)$
24 $\Sigma_a \leftarrow \left\{(\phi_{a,j}, d_{a,j}) | \Delta y_{a2}(j) \Delta y_{a2}(j+1) < 0 \cap \Theta_s\right\}$
25 Transmit Σ_a to all the requested robots $R_i \in \mathcal{R}_a$
26 **end if**/* Force calculation for state transition */
27 Calculate $\vec{F}_a, \vec{F}_{ai}$ from (8) and (13)
28 **return** $\Theta_s, \vec{F}_a, \vec{F}_{ai}$

A. Planning Process

The planning process based on Nav-SCOPE is a minimal reactive solution, which has proved to be versatile in a wide range of applications [3], [7], [8], [9]. A finite state machine (FSM) is illustrated in Fig. 5(c), where a robot R_a will go through a series of states $\mathcal{S}_a \in$ {i, ii, iii, iv} planning towards the target. Algorithm 1 illustrates the detailed process of Nav-SCOPE that obtains safe directions and interaction forces for state transition, where $\mathscr{F}_{N_f}[\cdot]$ indicates the FFT transform with N_f points, $\mathcal{N}_a$ is the group of observed neighbors, and $\mathcal{R}_a$ contains teammates that ask for environmental features. The information flow is formed only when needed.

During the planning process, the robot R_a needs to update safe direction set (Lines 3-6) and observed neighbor list (Lines 7-15) continuously. When the target direction is within safe directions, that is, $\vec{e}_a \in \Theta_s$, then R_a will advance straight towards the target ($\mathcal{S}_a$ = i). Otherwise, the target is blocked by an obstacle B, and R_a will request a cooperative perception ($\mathcal{S}_a$ = ii, Lines 16-18) combined with its own observation (Lines 22-24) to find an optimal direction *dir*. A local group will be naturally formed with the neighboring robots $\mathcal{N}_a$, where the environmental features are sent (Lines 8-9 and Line 25 at neighbors) and gathered at R_a. Then, the interaction forces are calculated (Line 27), and the probabilistic fusion is achieved by (15). Afterwards, R_a will search along the given *dir* towards a safe way closest to its target:

$$\min_{m \in dir} \{\langle \theta_m; \theta_{\vec{e}_a} \rangle\}, \text{ s.t. } \theta_m \in \Theta_s. \tag{16}$$

Subsequently, R_a will enter the obstacle following state ($\mathcal{S}_a$ = iii). In case there is a dead end, where $\Theta_s = \varnothing$, the robot will rotate in place towards *dir* while updating Θ_s, before it meets $\Theta_s \neq \varnothing$ and finds a way out. It then keeps a safety margin from the obstacle borders, until it circumnavigates the obstacle, where $\vec{e}_a \in \Theta_s$ satisfies again. Finally, it resumes to advance towards the target. The reactive strategy of Nav-SCOPE diverges from PF methods [7], [8], as robot paths are generated directly within safe regions. This feature gets rid of complicated computing for repulsive fields, and avoids saddle points commonly encountered in current methods. Compared with Bug algorithms [9], [10], Nav-SCOPE introduces cooperative perception and fusion optimization, which achieve much better paths and coordination.

In order to prevent path conflicts among robots, our model incorporates a collision avoidance state ($\mathcal{S}_a$ = iv). When a robot is subjected to any repulsive interaction $\vec{F}_{ai}$ from neighbors greater than its own propulsion, it then enters the collision avoidance state:

$$\exists_i, (\vec{F}_a + \vec{F}_{ai}) \cdot \vec{e}_a \leq 0. \tag{17}$$

In such circumstances, robot R_a will wait for its neighbor to pass until the above condition is no longer satisfied.

B. Mutual Positioning

In order to fuse cooperative perception within the same coordinate system, as well as calculate interaction forces, it is essential to determine relative poses and distances between neighboring robots. Instead of using external equipment, an engineering improvement is introduced to achieve mutual positioning in a distributed way. In this study, each robot is equipped with an ultra-wide band (UWB) beacon and four 90° directional array antennas with three units, as shown in Fig. 4(d). The range and azimuth of neighboring robots are decided by the angle of arrival (AOA) of the UWB beacons and the time difference of arrival (TDOA) between antenna units [29]. The relative poses of robots can be determined from geomagnetism. In real-world experiments, these are achieved by Nooploop NIUB01 IOT models and CMP10A electronic compasses.

C. Parameter and Variable Determination

The proposed Nav-SCOPE is interpretable, as its parameters are of explicit physical significance. We provide a way to assign parameters and variables without gaps for deployment.

1) Safety Margin r: It is determined by robot's max velocity v_m, max acceleration a_m, and the sensor delay t_d:

$$r = r_c + v_m t_d + \frac{v_m^2}{2a_m}, \tag{18}$$

where the first term refers to the radius of robot's circumcircle, the second term represents the distance s_d caused by the sensor delay, and the third term denotes the braking distance s_b.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

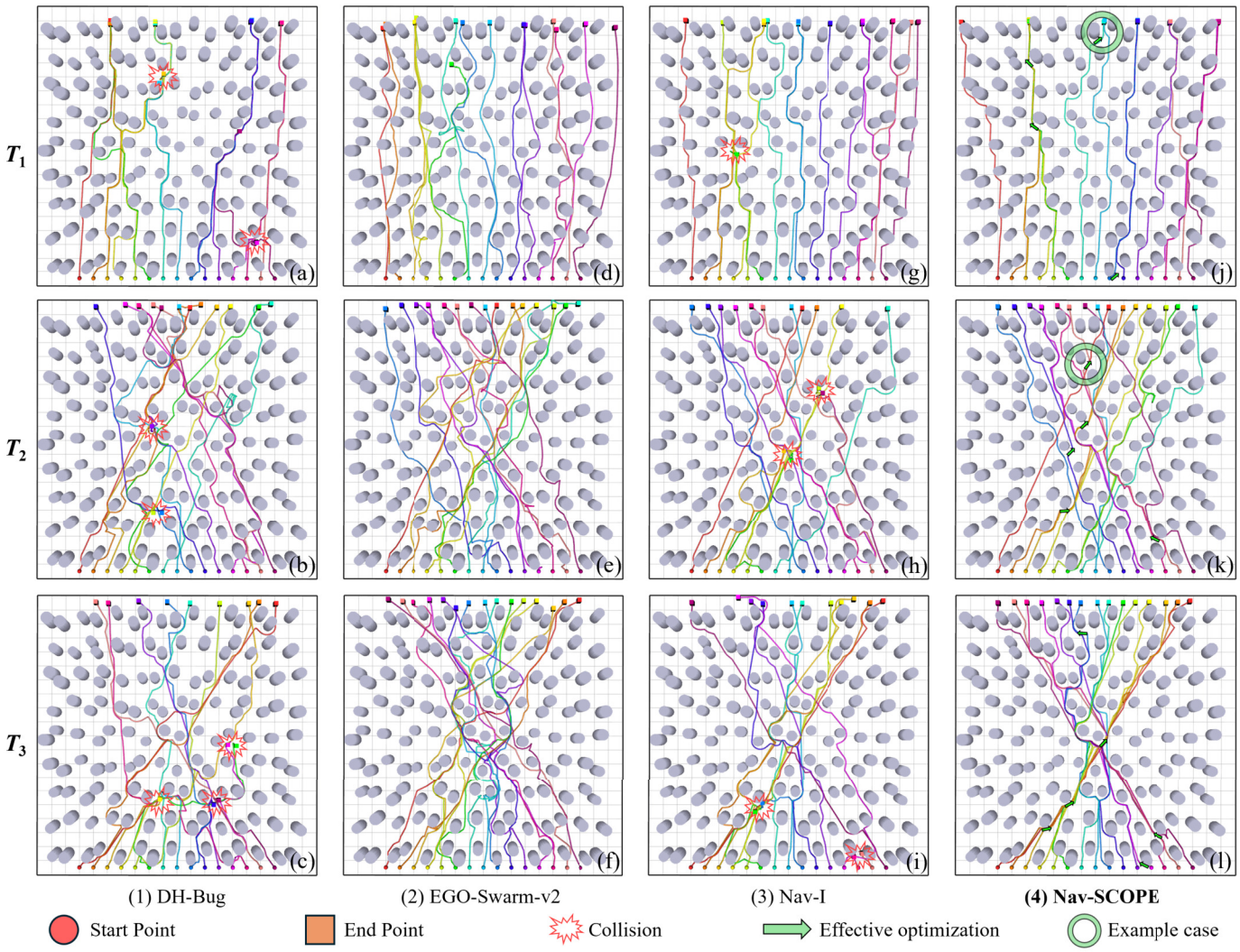

Fig. 6. Result comparison between Nav-SCOPE and baselines. Each row corresponds to a specific task, and the columns represent four algorithms.

2) Reference Distance σ_a: It is set to keep the distances between robots greater than $2r$ as follows:

$$\sigma_a = 2(r + s_d + s_b). \tag{19}$$

3) Distance Threshold $\mathbf{w}_n$: It distinguishes passable ways in neighbors' locations, where robots are expected to advance safely for a while towards the target. Hence, it is assigned as twice the total length of the protective model:

$$\mathbf{w}_{n,k} = 2(r + l_{th}) + s_d + s_b, k = 1, 2, \tag{20}$$

where k is the quadrant index. The value of $\mathbf{w}_n$ should be normalized to the same scale as the sensory data $s_a(n)$.

4) Interaction Force Weight w_i: It modulates the interaction amplitudes in $\vec{f_{ai}}$. The optimal value can be theoretically determined by AdaBoost [30] with error rates ϵ_s and ϵ_n of the corresponding probabilities P_a and P_{ni}. If these parameters are not available, they can be estimated with robust performance using extremely small samples. A sampling playground is randomly generated separate from the maps in the simulation, and nine samples of the swarm are used for an approximate estimation. This process is illustrated in Fig. 5(a), and the estimated performances are shown in Fig. 7(b). Consequently, we get $\epsilon_s = 0.245$ and $\epsilon_n = 0.446$, and the optimal amplitude of $\vec{f_{ai}}$ in (15) should be:

$$\|\vec{f_{ai}}\| = \frac{\frac{1}{2}\log\left(\frac{1-\epsilon_n}{\epsilon_n}\right)}{\frac{1}{2}\log\left(\frac{1-\epsilon_s}{\epsilon_s}\right)} = 0.195. \tag{21}$$

The value of w_i should be adjusted to approach the optimal value. We set $w_i = 1$ in experimental applications and use $w_i = 0.25, 0.5, 2.0, 4.0$ in simulations for comparison.

5) Self Observation $\mathbf{x}_0$: This vector records the estimated open space area from Q_1 to Q_4. Incorporating environmental features and linear interpolation described in Section IV-B, the elements of $\mathbf{x_0}$ correspond to the reconstructed area with each quadrant:

$$\begin{aligned} x_{0,k} &= \frac{1}{2}\int_{\phi_{k,l}}^{\phi_{k,u}} \left(\frac{d_{k,u} - d_{k,l}}{\phi_{k,u} - \phi_{k,l}}\phi + \frac{d_{k,l}\phi_{k,u} - d_{k,u}\phi_{k,l}}{\phi_{k,u} - \phi_{k,l}}\right)^2 d\phi \\ &= \frac{\phi_{k,u} - \phi_{k,l}}{6}\left(d_{k,l}^2 + d_{k,l}d_{k,u} + d_{k,u}^2\right), k = 1, \cdots, 4 \end{aligned} \tag{22}$$

where $\phi_{k,u}$ and $\phi_{k,l}$ are the upper and lower boundaries of the k-th quadrant, and $d_{k,u}, d_{k,l}$ are the corresponding distances.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

6) Neighbor Observation $\mathbf{x}_{ni}$*:* Its elements are estimates of passable distances described in (14). In order to identify open spaces with fewer obstacles, $\mathbf{x}_{ni}$ is assigned in a fuzzy manner:

$$x_{ni,k} = \begin{cases} (\max d_{i,k}, \min d_{i,k})/2 & \exists \text{ local minima} \\ \max d_{i,k} & \text{other cases,} \end{cases} \tag{23}$$

where $k = 1, 2$ for Q_1 and Q_2. Since local minima indicate potential obstacles, the minimum value delivers a message of potential obstruction. For the case of no local minima, the maxima value can effectively emphasize potential pathways.

VI. Simulation Results

In this section, Nav-SCOPE is applied to robot swarms in cluttered and unknown environments on Gazebo using an i9-12900K CPU. We compare the performance of Nav-SCOPE with 3 baselines: a current reactive solution DH-Bug [10], a current SOTA method EGO-Swarm-v2 [1], and an ablation baseline $w_i = 0$ referred to as Nav-I, concerning individual robots without neighboring scope to form the interaction fields.

To evaluate performances, we use the following statistical metrics: (a) Robot arrival rate (AR), indicating the ratio of arrived robots to the total number of swarm robots. (b) Path redundancy γ, referring to the redundant path relative to the optimal path length. Inspired from the information theory [31], γ is defined as

$$\gamma = 1 - AR \cdot \frac{\Sigma_i l_i^*}{\Sigma_i \bar{l}_i}, \tag{24}$$

where i is the robot index within a swarm, l^* signifies the optimal path length, and $\bar{l}$ denotes the average path length. Here, we use the path from A* [32] with a resolution of 0.05m as the optimal path. (c) Duration of the task t.

Swarms of 15 robots are tested in very dense forests, about 0.35 obstacles/m^2 density in a $400m^2$ region. The physical characters of the robot are $r_0 = 0.15$m, $v_m = 0.5$m/s, and $a_m = 2\text{m/s}^2$. The sensor we used is a 5Hz LiDAR with an observation range of $R = 5$m. The parameters are: $r = 0.30$m, $l_{th} = 0.60$m, $\alpha = 60°$, and $\sigma_a = 0.9$m. The maximum simulation time is set to 5 minutes. We compared performance in three tasks:

- T_1: Cooperative traversing from one side to the other.
- T_2: Intersected targets with different start and end points.
- T_3: Enforced collision avoidance with extreme interferences.

The statistical results for 20 random maps are detailed in Table I and Fig. 7(a). The optimal parameter w_i aligns with the sampled results in Section V-C, as illustrated in Fig. 7(b).

Fig. 6 depicts results from a random forest in three scenarios. Many redundant paths can be observed in three baselines, mainly caused by limited FOV, obstacle obstructions, and robot intersections. DH-Bug performs greedy reactive behaviors towards the target. But sometimes more haste, less speed. It often chooses worse directions with more detours and conflicts. EGO-Swarm-v2 impressively identifies local open spaces to generate long-term paths with much fewer circumnavigations. However, the restricted observation occasionally results in deviations and reversals. Nav-I performs suboptimal navigation without interaction fields from neighbors, making it almost degenerate into DH-Bug. In contrast, the performance of Nav-SCOPE is robust across different tasks. Coordinated navigation is achieved with the information flow from cooperative perception. Specifically, in the following aspects:

TABLE I
Statistical Results of Nav-SCOPE and Baselines

Task	Algorithm	γ (%)	AR (%)	t (s)
T_1	DH-Bug	16.3	90.3	88.0
	EGO-Swarm-v2	10.6	94.7	123.4
	Nav-I	13.1	92.7	81.8
	Nav-SCOPE	**5.1**	**100**	**66.2**
T_2	DH-Bug	27.6	79.7	125.5
	EGO-Swarm-v2	15.5	92.7	149.4
	Nav-I	30.2	75.7	132.9
	Nav-SCOPE	**4.4**	**99.7**	**85.0**
T_3	DH-Bug	34.1	75.0	140.4
	EGO-Swarm-v2	24.7	85.3	184.5
	Nav-I	35.8	71.3	145.2
	Nav-SCOPE	**7.1**	**98.7**	**99.0**

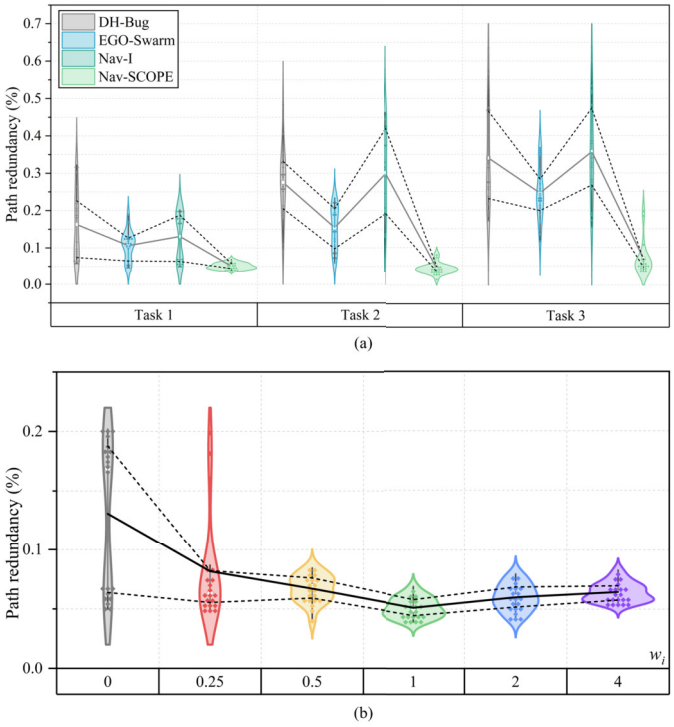

Fig. 7. (a) Path redundancy γ across different tasks and algorithms. (b) Path redundancy γ with different w_i.

A. Convergence Effects

The arrow markings in Fig. 6(j)-(l) show some cases where paths are optimized through spontaneous synergies within local teams. A detailed case is depicted in Fig. 4(a). When a robot identifies a safe gap with fewer obstacles, it attracts nearby neighbor robots to follow its path. As a result, local teams are naturally formed by convergence effects along paths with relatively fewer obstacles. This type of effect is absent

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

TABLE II
COMPUTATION AND COMMUNICATION USAGE

	t_{plan} (μs)	Memory (KB)	Bandwidth[1] (Kbps)	Package Size (Byte)
DH-Bug	1.8	1.6	/	/
Nav-I	15.8	14.8	/	/
EGO-Swarm-v2	1.2×10^3	3.5×10^3	316.5	289.9
Nav-SCOPE	15.9	14.9	33.6	80.8

[1] Syetem level.

from the baseline methods. Specifically, the ablation study of Nav-I indicates that optimization with local perception alone has limited impacts on long-term performance, particularly in complicated tasks.

B. Divergence Effects

Note that the information from the cooperative perception may be negative when neighbors detect obstacles. Fig. 4(b) illustrates a case when the robot R_b in front encountered an obstacle, the following robot R_a turned to the other side. Thus, robots can avoid unnecessary detours caused by perceptual limitations and optimize future paths with fused observation. EGO-Swarm-v2 also exhibited divergence effects as it tends to generate scattered paths to avoid collisions. But this function sometimes leads to unnecessary detours.

C. Collision Avoidance

This feature is achieved by the repulsive effect of the interaction field, where planning directions are considered to prevent future interference between neighboring robots. Simulation results demonstrate safe and organized paths, especially when robots are enforced to avoid reciprocal collisions, as shown in Fig. 6(l). However, for EGO-Swarm-v2, the loss of DOF on the Z axis makes it challenging to find a collision-free path for mobile robots in dense situations. Hence, planner timeouts and aggressive trajectories occasionally occurred, resulting in robot failures and longer task durations.

D. Resource Consumption

The statistical performances of computation and communication are summarized in Table II, where t_{plan} denotes the end-to-end time delay from perception to planning execution. In terms of computation, Nav-SCOPE performs the same order as minimal reactive methods and is much lower compared to EGO-Swarm-v2. The computation mainly comes from distributed FFT filtering and feature extraction on each robot. Compared with Nav-I, we can see that the calculation of interaction fields costs only 0.1 μs on average. This is achieved by our ad-hoc connectivity design, which reduces the complexity of scaling to linear order. As for communication, Nav-SCOPE consumes a minimal bandwidth while carrying more information from cooperative perception. This is realized by compressed features in (7) and on-demand transmission.

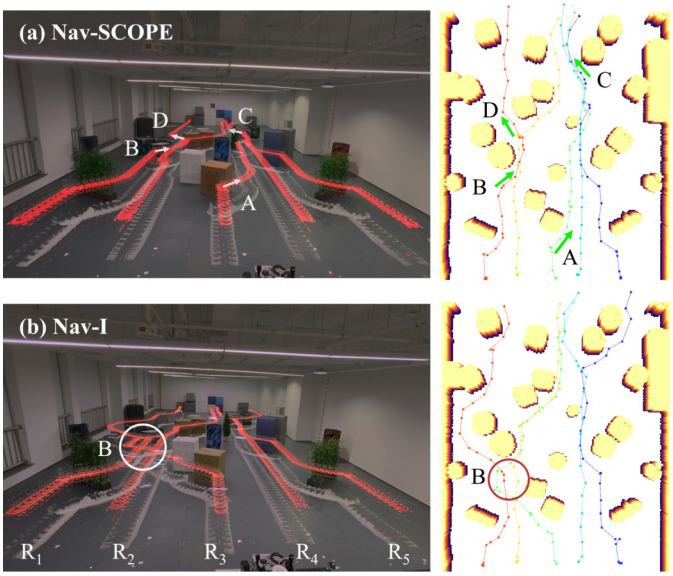

Fig. 8. Experimental results in a cluttered depository. To present the obstacles and trajectories with clarity, a point cloud map was produced *after* the experiments finished.

Consequently, Nav-SCOPE is an ultralight solution, suitable for deployment on robots with limited computational capacity or in scenarios with restricted communication.

VII. REAL WORLD EXPERIMENTS

Besides simulations, we also performed real-world experiments with a swarm of 5 robots. Each robot is equipped with an 8-core ARM v8.2 CPU and a 20Hz OS1 LiDAR with an effective range of 5m. Data transmission is performed by a 78.125 Kbps ad-hoc UWB network. The robots use Nooploop NIUB01 IOT models and CMP10A electronic compasses for mutual positioning. The robot characters are: r_0 = 0.29m, r_e = 0.43m, v_m = 0.5m/s, a_m = 4m/s^2. Consequently, we set r = 0.49m, l_{th} = 0.83m, α = 72° and σ_a = 1.09m. The LiDAR observation is represented as a polar coordinate array with approximately 360 points.

The experimental setup consists of a cluttered depository containing various shaped obstacles. The density is approximately 0.38 obstacles/m^2. The result is illustrated in Fig. 8. Our proposed Nav-SCOPE generates coordinated and collision-free trajectories in contrast to the individual Nav-I. The average path length of Nav-SCOPE is 10.77m with approximate γ = 3.8%, while Nav-I achieves 11.29m with much higher redundancy γ = 8.3%.

Specifically, we observe convergence effects at points A, B and C, and a divergence effect at point D. All of these are accomplished by incorporating supplementary information from cooperative perception. A particular case at point A is presented in Figure 9, where it is difficult for R_3 to decide its future direction, as the current observation looks similar on either side. But after fusion, it chose to follow R_4 rightwards, and there proved to be fewer obstacles. The swarm also gains advantages from collision avoidance by eliminating path interferences which evidently occurred at points D in the Nav-I

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

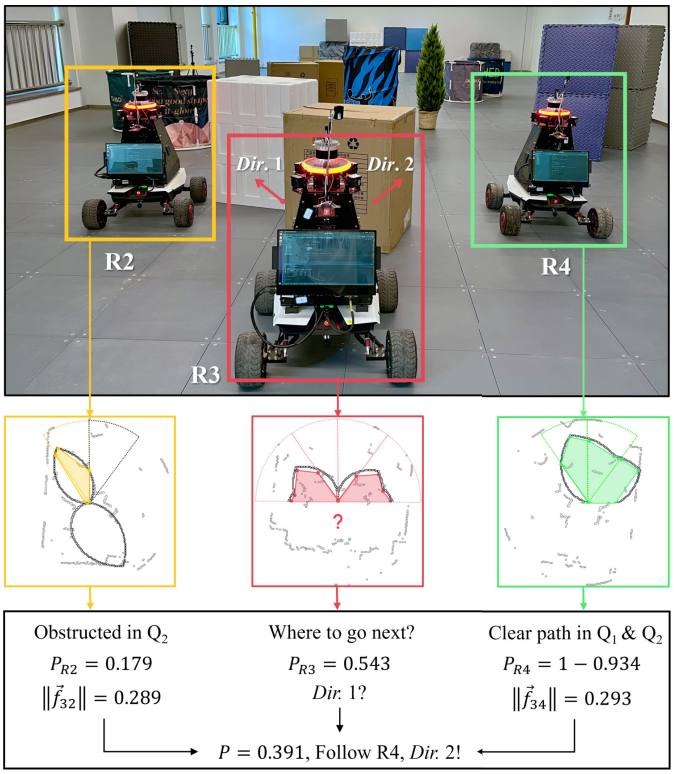

Fig. 9. An effective case of coordinated navigation within a local team.

baseline. In contrast, Nav-SCOPE shows organized trajectories despite robot interactions and narrow pathways.

VIII. Discussion and Conclusion

Despite the current success, we acknowledge that Nav-SCOPE is only applicable to ground robots at this stage, while extension is considered for applications in 3D space. Our future aim is to implement Nav-SCOPE on aerial robots, such as quadrotors. In the perception stage, we will introduce 2D FFT filtering to process depth images [33]. In the planning stage, we will extend the probability function from bi-categories to multi-categories, and incorporate trajectory optimization with robot dynamic costs [34]. In execution, we will add the time-delay calibration in high-speed scenarios [20].

Other promising areas include model adaptation to dynamic environments and task expansion to exploration applications. Some existing theories such as obstacle tracking [35] and target allocation [36], etc. could be incorporated to improve system performance in specific tasks.

In summary, we propose a systematic solution to achieve cooperative perception and coordinated navigation on swarm robots. Our approach facilitates information flow for observation fusion, followed by path optimization with self-organized convergence, divergence, and collision avoidance. Baseline comparisons demonstrate its exceptional performance, robustness, and lightweight demands.

References

[1] X. Zhou et al., "Swarm of micro flying robots in the wild," *Sci. Robot.*, vol. 7, no. 66, p. 17, May 2022.
[2] H. Wu, B. Tao, Z. Gong, Z. Yin, and H. Ding, "A standalone RFID-based mobile robot navigation method using single passive tag," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1529–1537, Oct. 2021.
[3] K. N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Sci. Robot.*, vol. 4, no. 35, p. 14, Oct. 2019.
[4] X. Wang, W. Liu, Q. Wu, and S. Li, "A modular optimal formation control scheme of multiagent systems with application to multiple mobile robots," *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9331–9341, Sep. 2022.
[5] G. Wang, X. Wang, and S. Li, "A guidance module based formation control scheme for multi-mobile robot systems with collision avoidance," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 1, pp. 382–393, Jan. 2024.
[6] L. Chen et al., "STR: Spatial–temporal RetNet for distributed multi-robot navigation," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 10429–10441, 2025.
[7] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 1985, pp. 500–505.
[8] E. Soria, F. Schiano, and D. Floreano, "Predictive control of aerial swarms in cluttered environments," *Nature Mach. Intell.*, vol. 3, no. 6, pp. 545–554, May 2021.
[9] V. Lumelsky and A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Trans. Autom. Control*, vol. AC-31, no. 11, pp. 1058–1063, Nov. 1986.
[10] Y. Zhu, T. Zhang, J. Song, and X. Li, "A new bug-type navigation algorithm for mobile robots in unknown environments containing moving obstacles," *Ind. Robot, Int. J.*, vol. 39, no. 1, pp. 27–39, Jan. 2012.
[11] 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.
[12] D. Duberg and P. Jensfelt, "UFOExplorer: Fast and scalable sampling-based exploration with a graph-based planning structure," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2487–2494, Apr. 2022.
[13] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, Aug. 2023.
[14] J. Lu, X. Zhang, H. Shen, L. Xu, and B. Tian, "You only plan once: A learning-based one-stage planner with guidance learning," *IEEE Robot. Autom. Lett.*, vol. 9, no. 7, pp. 6083–6090, Jul. 2024.
[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Conf. Artif. Intell.*, 2002, pp. 593–598.
[16] J. Xiao, P. Pisutsin, and M. Feroskhan, "Collaborative target search with a visual drone swarm: An adaptive curriculum embedded multistage reinforcement learning approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 313–327, Jan. 2025.
[17] B. Zhou, H. Xu, and S. Shen, "RACER: Rapid collaborative exploration with a decentralized multi-UAV system," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1816–1835, Jun. 2023.
[18] W. Lu, T. Zhang, J. Yang, and W. Xueqian, "A formation control approach with autonomous navigation of multi-robot system in unknown environment," in *Proc. 34th Chin. Control Conf.*, 2015, pp. 5230–5234.
[19] M. Doostmohammadian, A. Taghieh, and H. Zarrabi, "Distributed estimation approach for tracking a mobile target via formation of UAVs," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3765–3776, Oct. 2022.
[20] M. Doostmohammadian, M. Pirani, and U. A. Khan, "Consensus-based networked tracking in presence of heterogeneous time-delays," in *Proc. 10th RSI Int. Conf. Robot. Mechatronics (ICRoM)*, Nov. 2022, pp. 17–22.
[21] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "EGO-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Oct. 2021, pp. 4101–4107.
[22] L. Chen et al., "Reciprocal velocity obstacle spatial–temporal network for distributed multirobot navigation," *IEEE Trans. Ind. Electron.*, vol. 71, no. 11, pp. 14470–14480, Nov. 2024.
[23] Y. Zhou, J. Xiao, Y. Zhou, and G. Loianno, "Multi-robot collaborative perception with graph neural networks," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2289–2296, Apr. 2022.
[24] M. Goarin and G. Loianno, "Graph neural network for decentralized multi-robot goal assignment," *IEEE Robot. Autom. Lett.*, vol. 9, no. 5, pp. 4051–4058, May 2024.
[25] W. Lu, Q. Lin, L. Meng, C. Li, and B. Liang, "Collaborative navigation of multiple autonomous mobile robots via geometric graph neural network," *IEEE Trans. Ind. Electron.*, early access, Jun. 19, 2025, doi: 10.1109/TIE.2025.3574516.

[26] C. Li, W. Lu, Z. Ma, L. Meng, and B. Liang, "Highly efficient observation process based on FFT filtering for robot swarm collaborative navigation in unknown Environments*," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Abu Dhabi, United Arab Emirates, Oct. 2024, pp. 10267–10274.
[27] J. Kaiser and R. Schafer, "On the use of the I0-sinh window for spectrum analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 1, pp. 105–107, Feb. 1980.
[28] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, pp. 4282–4286, May 1995.
[29] *IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYS) and Associated Ranging Techniques*, Standard 802.15.4z-2020, 2020, pp. 1–174.
[30] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[31] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[32] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
[33] R. M. Mersereau and D. E. Dudgeon, "Two-dimensional digital filtering," *Proc. IEEE*, vol. 63, no. 4, pp. 610–623, 1975.
[34] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1294–1310, Dec. 2015.
[35] T. Eppenberger, G. Cesari, M. Dymczyk, R. Siegwart, and R. Dubé, "Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, USA, Oct. 2020, pp. 10528–10535.
[36] B. Cheng, M. He, Z. Zhu, B. He, and J. Chen, "Development and application of coverage control algorithms: A concise review," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 14906–14927, 2025.

Chenxi Li (Graduate Student Member, IEEE) received the B.E. degree in communication engineering from the College of Electronic and Information Engineering, Tongji University, Shanghai, China, in 2022. He is currently pursuing the Ph.D. degree in control science and engineering with the Department of Automation, Tsinghua University, Beijing, China.

His research interests mainly focus on robotics, including perception, planning, and embodied intelligence.

Weining Lu received the B.S. degree in physics from the Department of Physics, Fudan University, Shanghai, China, in 2011, and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2017.

He is currently an Assistant Research Fellow with Beijing National Research Center for Information Science and Technology, Tsinghua University. His current research interests include multi-agent systems, computer vision, and data mining.

Qingquan Lin received the B.E. degree in mechatronics engineering from Zhejiang University, Hangzhou, China, in 2019, and the M.S.E. degree in control science and engineering from Tsinghua University, Beijing, China, in 2022.

Currently, he is a Research Assistant with the Department of Automation, Tsinghua University. His research interests revolve around robot perception and collaborative planning.

Litong Meng (Member, IEEE) received the B.E. degree in robotics engineering from the College of Information Science and Electrical Engineering, Shandong Jiaotong University, Shandong, China, in 2022.

He is an Engineer with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include multi-robot systems planning and control.

Haolu Li received the B.E. degree in mechatronic engineering from the School of Mechanical and Electrical Engineering, Beijing Information Science and Technology University (BISTU), Beijing, China, in 2022, and the M.S. degree in mechanical engineering from BISTU in 2025.

His current research interests include machine vision, intelligent perception, and control of robots.

Bin Liang (Senior Member, IEEE) received the B.S. and M.S. degrees in control engineering from the Honors College, Northwestern Polytechnical University, Xi'an, China, in 1989 and 1991, respectively, and the Ph.D. degree in control engineering from the Department of Precision Instrument, Tsinghua University, Beijing, China, in 1994.

From 1994 to 2007, he held his positions as a Post-Doctoral Researcher, an Associate Professor-Level Researcher, a Professor, and an Assistant Chief Engineer with China Aerospace Science and Technology Corporation, Beijing. Since 2007, he has been a Professor with the Department of Automation, Tsinghua University. His research interests include modeling and control of robotic systems.