

RE-Formation: Resilient and Efficient Formation Planning in Large-Scale Distributed Aerial Swarms

Yuan Zhou, Lun Quan, Chao Xu, Guangtong Xu, and Fei Gao

Abstract—Due to the limited online computational resources and the inherent probability of hardware and software failures of real-world robots, large-scale formation planning faces two common challenges: computational intractability and agent failures. Based on the theory of sparse graphs and the maximum clique, we achieve a resilient and efficient formation planning (RE-Formation) to address these issues. To improve the computational efficiency of trajectory planning while ensuring flexible formation maneuvers, we introduce sparse graphs to describe connection relationships and present a sparse graph construction method with closed-form solutions. The sparse graphs ensure the *Global Rigidity* for uniquely corresponding to a geometric shape and *Preserve the main Features* of complete graphs, denoted as the GRPF sparse graph. To prevent the impact of abnormal agents, the problem of eliminating abnormal agents is transformed into an outlier rejection problem that can be solved by computing the maximum clique. We approximate the maximum clique by periodically triggering the calculation of the maximum k -core to meet the real-time computational demands of large-scale swarms. We validate the performance through real-world experiments and implement formation planning with 100 drones in simulation. Benchmark comparisons and ablation experiments demonstrate the effectiveness of our method.

Note to Practitioners—In this paper, we address the challenges of large-scale aerial swarm formation planning, focusing on computational efficiency and resilience to agent failures. Real-world swarm robots often face limitations in computational resources and are prone to hardware and software failures, which can severely impact formation coordination. Our proposed method uses sparse graphs that are globally rigid to reduce computational complexity while ensuring flexible formation maneuvers, and employs outliers rejection to eliminate abnormal agents, ensuring resilient collaboration.

Index Terms—Large-scale formation, resilient swarms, efficient planning, sparse graphs, outlier rejection.

I. INTRODUCTION

Formation planning is a fundamental requirement for swarm robots. Robots in real-world scenarios are inherently imperfect [1], [2], often limited by computational resources and a non-negligible probability of hardware and software

This work was supported by the National Natural Science Foundation of China under Grant 62322314 and Grant 62203256. (*Corresponding Author:* Guangtong Xu; Fei Gao.)

Yuan Zhou, Lun Quan, Chao Xu, and Fei Gao are with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute, Zhejiang University, Huzhou 313000, China. (e-mail: {y2zhou, lunquan, cxu, fgaoaa}@zju.edu.cn). Guangtong Xu is with the Huzhou Institute, Zhejiang University, Huzhou 313000, China. (e-mail: guangtong_xu@163.com).

Guangtong Xu is with the Huzhou Institute, Zhejiang University, Huzhou 313000, China. (e-mail: guangtong_xu@163.com)

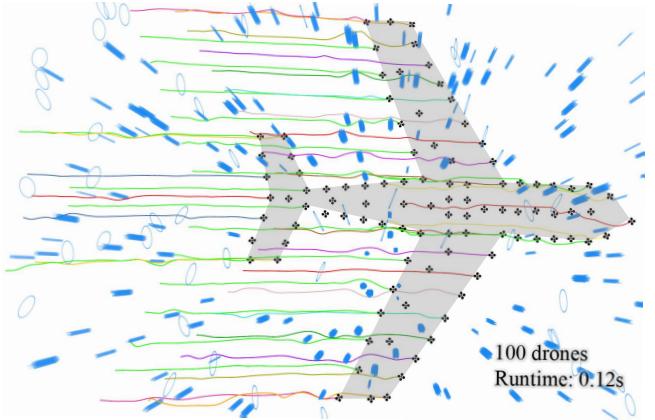


Fig. 1. Simulation results of the resilient and efficient formation planning that maintaining the shape of a jet aircraft with 100 drones while avoiding obstacles.

failures. As the scale of formation increases, the additional constraints imposed by the collaborative relationships among agents challenge the finite computational resources. Moreover, abnormal status information induced by agent failures, which is prone to occur in large-scale swarms, can propagate through the collaborative network to other agents and significantly impact the formation coordination of the remaining agents. Therefore, a reasonable cooperation relationship becomes the key to ensuring the performance of the large-scale formation.

Numerous works have demonstrated various cooperative relationships. Intuitive methods include collaboration that only considers information from neighboring agents. It is simple and effective that agents do not need to process and rely on the information from all agents, which facilitates the formation of extensive swarms, and failure of agents does not significantly affect overall cooperation [3], [4]. However, such swarm systems suffer from protracted recovery time to reestablish the desired formation when responding to disturbances. Some works use density to abstractly represent the overall formation state and still struggle to adequately characterize the entire formation, making it hard to respond to disturbances quickly [5], [6]. Conversely, a distinct category of methods utilizes the state information of all other agents [7], [8]. Agents can assimilate the overall state of the swarm, enabling coordinated navigation through complex environments while maintaining formation stability and achieve timely recovery from disruptions. Nonetheless, the entire formation becomes highly vulnerable to the disproportionate impact of a few failed agents under a fully-connected collaborative network, while typically requires considerable computational resources as the increasing constraints lead

to an extended solving time when the scale escalates. In summary, current formation systems struggle to simultaneously achieve large-scale swarm motion planning in complex environments and tolerance to agent failures.

In nature, individuals within a flock of birds only need to consider the movements of a few individuals to cooperate, allowing them to form and maintain large-scale flocking [9]. Even if some individuals stray from the flock, it does not significantly affect overall cooperation [10]. The objective of this paper is to establish a resilient and large-scale formation navigation system in complex environments like a flock of birds based on sparse collaborative relationships. To achieve obstacle avoidance and formation keeping while ensuring flexible maneuvers, we adopt the framework of distributed trajectory optimization [1], and employ graphs to measure formation similarity. The overall framework of our algorithm flow is shown in Fig. 2. To reduce the excessive constraints caused by large-scale swarms, we use the sparse graph to replace the computationally demanding complete graph, which means that agents only need to consider the information from a subset of other agents. After determining the sparsification rate based on the actual ability of the computation resource on board, we transform the construction of a sparse graph into the problem of submatrix selection [11] of the Laplacian matrix. To prevent the abnormal state information of failed agents from affecting the majority, we eliminate edges with significant disparities and compute the maximum clique that minimizes the error between the graph of desired formation and the current formation. By optimizing trajectories with the normal agent states in the maximum clique and applying the sparse graph as formation constraints, we achieve Resilient and Efficient larger-scale **Formation** called **RE-Formation**. The ablation experiments demonstrate the resilience of our abnormal state elimination method in formation flight. Physical experiment and extensive benchmarks verify the efficiency and performance superiority of our sparse graph for differentiable metric in formation planning. Compared to the cutting-edge distributed formation planning method [1], planning efficiency is improved by approximately an order of magnitude when the number of drones exceeds 80. Moreover, we achieve the real-time formation planning of 100 agents in a complex environment through simulation as shown in Fig. 1. The contributions of this paper are as follows:

- 1) We propose a **GRPF** sparse graph construction method for formation planning by submatrix selection. We explore metrics in submatrix selection related to achieving formation performance, with the quantification of Max-Trace as the optimal metric. Moreover, a closed-form solution is derived utilizing Max-Trace, enabling the real-time construction of a **GRPF** sparse graph.
- 2) We analysis that the **GRPF** sparse graph is globally rigid for uniquely corresponding to a formation shape, and preserve the main structural feature of the complete graph to guarantee formation performance. Through computational complexity analysis and experimental

validation, **GRPF** sparse graph exhibits superior computational efficiency in formation planning.

- 3) We transform the agent failure tolerance into the problem of rejecting outliers and solve it by computing the maximum clique. Furthermore, we compute the maximum k -core to identify the approximation of the maximum clique efficiently due to its linear complexity [12]. By incorporating the method of eliminating abnormal state information into formation trajectory planning, we achieve a resilient formation.

II. RELATED WORKS

Numerous research focus on developing swarm and formation systems [1], [3], [13], [14]. We conduct a review of existing research on large-scale formation generation. In addition, a survey of the existing literature on formation navigation is conducted in this paper. Furthermore, we extend the investigation to resilience enhancement strategies, particularly those incorporating agent failures.

A. Lager-scale Formation Generation

Achieving large-scale swarm coordination has always been one of the research hotspots. An strategy formulates the target configuration as a set of probability distributed bins, utilizing the Markov chain algorithm to converge toward the specified formation [5], [15], [16]. Bandyopadhyay et al. [5] propose a probabilistic swarm guidance algorithm using inhomogeneous Markov chains, which controls the density distribution of autonomous agents through real-time feedback, achieving the desired formation shape with simulations involving up to one million agents. However, the slow convergence to desired configurations and reliance on obstacle-free environments render it inadequate for complex scenarios. Artificial potential fields (APF) are used to define target geometries [3], [6], [17], guiding agents through APF attractive and swarm repulsive forces to achieve formation generation. Nagpal et al. [3] utilize robots with a simple structure, achieving the self-assembly of 2D shapes through programmable local interactions and local sensing. Zhao et al. [6] develop a shape assembly strategy for robot swarms based on the mean-shift algorithm, similarly based on density control methods, enabling the swarms to form complex shapes with high adaptability. However, these approaches exhibit inefficient convergence, lack explicit obstacle avoidance, and fail to meet real-time planning demands in complex environments.

B. Formation Navigation in Complex Environments

There are many outstanding works on formation navigation in complex environments, including virtual structures [18], [19], consensus-based control laws [20], [21], leader-follower [22], [23], which design compliant control laws to achieve formation maintenance and obstacle avoidance. These methods typically require low computational resources. To further reduce communication overhead and simplify constraints, sparse graph-based communication architectures are adopted [4], [24], [25]. Moreover, ensuring

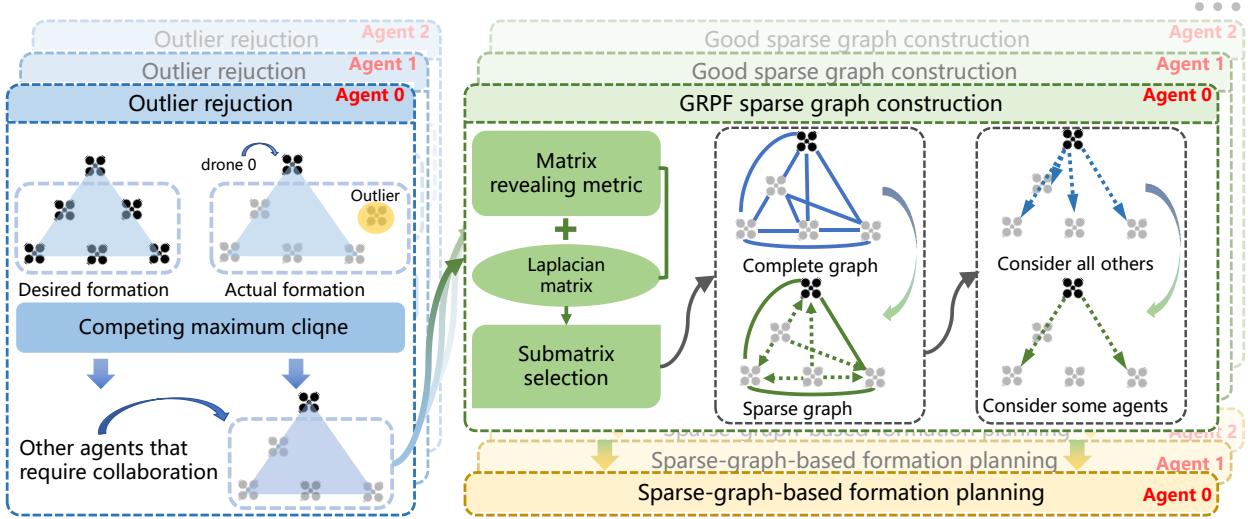


Fig. 2. The overall framework of our proposed planning method. The blue dashed box on the left illustrates the method for excluding anomalous failed agents, where each agent independently eliminates the outliers within the yellow background box. The green dashed box in the upper right indicates the connection relationships established in a sparse graph after removing the abnormal agents from the swarm, thereby further reducing the number of agents required for collaboration. The solid lines between agents represent bidirectional connections, while the dashed lines indicate unidirectional connections, and the arrows denote the direction of the connections. The yellow dashed box in the lower right signifies that only a subset of agents is considered for formation trajectory planning.

convergence in formation control requires maintaining the rigidity of the communication topology [26], [27]. Theoretical analysis reveals that the convergence rate is intrinsically correlated with the second smallest eigenvalue of the Laplacian matrix of the graph [28]. However, these local feedback methods exhibit limitations in complex obstacle environments and are prone to deadlock.

In contrast, optimization-based predictive frameworks strike a balance between obstacle avoidance and formation maintenance. However, such approaches require substantial computational resources. While centralized formation planning achieves superior collective mobility [7], [29], simultaneous trajectory optimization for all agents becomes computationally prohibitive for large-scale real-time applications. Treating the formation as a single entity to search for traversable regions in the map can reduce computational complexity. Nevertheless, these methods either rely on pre-processed maps [30], making real-time planning challenging, or simply employ flight corridors to characterize navigable areas [14], [31], resulting in inadequate adaptability in complex environments. Another category of methods adopts the framework of distributed trajectory optimization [1], [8]. However, formation planning based on complete graphs introduces excessive constraints, resulting in prohibitive computational costs and limited scalability. Some work successfully employ reinforcement learning to achieve formation maintenance in obstacle environments [32], [33], [34]. Xie et al. [33] achieve static and dynamic obstacle avoidance for formation with three agents. Compared with optimization-based formation planning methods, it has a stronger formation maintenance capability. However, such methods fail to achieve large-scale formation coordination in complex environments. Besides, the approaches described above rarely address the impact of the anomalous state generated by failed agents on formation.

To deploy large-scale formation planning systems in complex environments, this paper employs the distributed trajectory planning framework. A sparse graph preserving the spectral features of the original complete graph is constructed by submatrix selection [35], [36].

C. Agents Failures Tolerance and Outliers Rejection

Numerous studies have concentrated on the resilience of swarms, with particular emphasis on the critical task of addressing the resilience of swarms in the context of robot dropouts [37], [38]. However, those works focus on failures at the communication network level of robots and make several assumptions which limit practical applications. Some works consider robot failures at task assignment, Morgan et al. [39] determine the number of remaining agents in a swarm based on bidding information, while allocating target positions according to the number of bids. In this paper, we focus on the impact of abnormal state information generated by failed agents on formation coordination. These abnormal state information can be regarded as outliers in formation trajectory optimization. We transform agent failure tolerance into the problem of rejecting outliers [40] and eliminate abnormal state information in trajectory optimization. There are many methods used to solve the problem of outliers rejection. One category of methods treats it as a maximum consensus problem (MC). It can be solved by Random sample consensus (RANSAC) [41] and by completing the maximum clique [42]. Another solution is M-estimation [43], which attaches a robust loss function to the original optimization cost. The commonly used method to address this issue is through Graduated Non-Convexity (GNC) [44]. Additionally, it can be relaxed to a semidefinite program (SDP) [45]. In our problem, we focus primarily on identifying outliers and then performing graph sparsification after removing the abnormal agents. To avoid affecting the subsequent sparsification

process and meet the real-time requirements of large-scale formation planning, we treat it as MC and approximate the maximum clique by computing the maximum k -core that has linear complexity.

III. RIGID GRAPH AND SUBMATRIX SELECTION

In this section, we introduce the concept of graph rigidity and discuss the basic principles of submatrix selection.

A. Globally Rigid Graph and Laplacian Matrix

The formation constraint of N drones is described by a directed graph $\mathcal{G} = (\mathbf{v}, \mathbf{e})$, where $\mathbf{v} := 1, 2, \dots, N$ and $\mathbf{e} \subset \mathbf{v} \times \mathbf{v}$ represent the set of vertices and edges, respectively. In directed graph \mathcal{G} , the vertex $v_i \in \mathbf{v}$ denotes the i -th drone with position vector $p_i = [x_i, y_i, z_i]^T$, and define the composite vector $\mathbf{P} = (p_1, \dots, p_N) \in \mathbb{R}^{3N}$. We define the rigidity function of the graph \mathcal{G} as the function

$$f_{\mathcal{G}} := (\dots, \|p_i - p_j\|_2, \dots), \quad (1)$$

where the $\|\cdot\|_2$ denotes the 2-norm and the $\|p_i - p_j\|_2$ is the length of the edge $e_{ij} \in \mathbf{e}$ connecting the vertex v_i and v_j , representing that drone i consider the geometric distance and trajectory information from drone j to cooperate. In this paper, we only use the Euclidean distance between agents as information for formation coordination. Unlike bearing-based [27] and displacement-based methods [46], rigidity analysis is conducted through the framework of bar-and-joint [26]. We adopt the following definitions of rigidity, which are taken from Asimow and Roth [47].

Definition 1 (Rigid graph): Let \mathcal{G} be a graph with \mathbf{v} . \mathcal{G} is rigid if there exists a neighborhood $\mathbf{U} \in \mathbb{R}^{3N}$ of \mathbf{P} such that $f_{\mathcal{G}}^{-1}(f_{\mathcal{G}}(\mathbf{P})) \cap \mathbf{U} = f_{\mathcal{K}}^{-1}(f_{\mathcal{K}}(\mathbf{P})) \cap \mathbf{U}$, where \mathcal{K} is the complete graph with the same \mathbf{v} as \mathcal{G} . This means that the graph will not undergo continuous deformation under disturbances.

Definition 2 (Globally rigid graph): The graph \mathcal{G} is globally rigid if $f_{\mathcal{G}}^{-1}(f_{\mathcal{G}}(\mathbf{P})) = f_{\mathcal{K}}^{-1}(f_{\mathcal{K}}(\mathbf{P}))$. That is, this graph uniquely identifies a formation shape.

The level set $f_{\mathcal{G}}^{-1}(f_{\mathcal{G}}(\mathbf{P}))$ consists of all possible points with edges of the same length. The set $f_{\mathcal{K}}^{-1}(f_{\mathcal{K}}(\mathbf{P}))$ of the complete graph \mathcal{K} consists of points related to rigid body motions. According to the definition in [48], a globally rigid graph not only guarantees that the corresponding geometric shape does not undergo continuous deformation under disturbances, but also uniquely determines a shape. Therefore, the necessary condition for graph-based formation planning is that the graph \mathcal{G} is globally rigid. Fig. 3 shows a graphical interpretation of graph rigidity.

Since the Laplacian matrix \mathbf{L} contains the structure of the graphs and reflects the connection relationships of vertices [49], we use \mathbf{L} to formulate the formation constraint.

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad (2)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{D} \in \mathbb{R}^{N \times N}$ denote the adjacency matrix and the degree matrix of the graphs, respectively. The

elements in \mathbf{A} and \mathbf{D} are obtained by

$$A_{i,j} = \begin{cases} \|p_i - p_j\|_2, & \text{if } e_{ij} \in \mathbf{e}, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

$$D_{i,j} = \begin{cases} \sum_{j=1}^N A_{i,j}, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

B. Graph Sparsification and Submatrix Selection

For a complete graph $\mathcal{G}^{cmp} = (\mathbf{v}^{cmp}, \mathbf{e}^{cmp})$, we propose an algorithm to construct a corresponding sparse graph $\mathcal{G}^{spr} = (\mathbf{v}^{spr}, \mathbf{e}^{spr})$ that is also globally rigid, where $\mathbf{v}^{spr} = \mathbf{v}^{cmp} := 1, 2, \dots, N$ and $\mathbf{e}^{spr} \subset \mathbf{e}^{cmp}$.

Given a set of vertices $\mathbf{v} := \{1, 2, \dots, N\}$, a sparse graph \mathcal{G}^{spr} that satisfies the global rigidity can be constructed following the Algorithm 1 (A1). When the number of swarms is less than 4, we directly construct a complete graph to maintain the formation (A1: lines 3-7), as there are few formation constraints. The core of A1 is the selection of key vertices, where the pseudocode of the function **SelectionVbas** is detailed in Algorithm 2 (A2). Due to distinct sparse graph conditions to achieve global rigidity in linear, planar, and solid formation shapes, we process these cases separately to obtain \mathbf{v}^{bas} , and construct a globally rigid sparse graph as demonstrated in A2.

1) **Linear Formations:** We directly select the k vertices as \mathbf{v}^{bas} through the function **GRPFVertexSelset** (A2: lines 4 & 5). The details of function **GRPFVertexSelset** can be found in section IV.

2) **Planar Formations:** We select k vertices as \mathbf{v}^{bas} through the function **GRPFVertexSelset** (A2: lines 6 & 7). Then, we introduce the following evaluation criteria [50]:

$$\exists v_a, v_b, v_c \subseteq \mathbf{v}^{cmp}, \det([\overrightarrow{v_b v_a}, \overrightarrow{v_c v_a}]^T) \neq 0, \quad (5)$$

to determine whether vertices are collinear, i.e., the principle of the function **IsCollinear** (A2: line 8), where $\det(\cdot)$ is the determinant. If these k vertices are collinear, we randomly select one vertex from the remaining vertices that is not collinear with \mathbf{v}^{bas} and add it to \mathbf{v}^{bas} (A2: lines 9-11).

3) **Solid Formations:** We select k vertices as \mathbf{v}^{bas} through the function **GRPFVertexSelset** (A2: lines 12 & 13). We introduce evaluation criteria [50]:

$$\exists v_a, v_b, v_c, v_d \subseteq \mathbf{v}^{cmp}, \det([\overrightarrow{v_b v_a}, \overrightarrow{v_c v_a}, \overrightarrow{v_d v_a}]^T) \neq 0, \quad (6)$$

to determine whether vertices are coplanar, i.e., the principle of the function **IsCoplanar** (A2: line 13). If these k vertices are coplanar, we randomly select one vertex from the remaining vertices that is not coplanar with \mathbf{v}^{bas} and add it to \mathbf{v}^{bas} (A2: lines 15-17).

Then, we use \mathbf{v}^{bas} to construct a complete graph $\mathcal{G}^{bas} = (\mathbf{v}^{bas}, \mathbf{e}^{bas})$, and set the remaining vertices as $\mathbf{v}^{rnn} = \mathbf{v} - \mathbf{v}^{bas}$ (A1: lines 10-12). Connect each vertex in \mathbf{v}^{rnn} to all vertices in \mathbf{v}^{bas} , but the vertices in \mathbf{v}^{rnn} do not connect

with each other (A1: lines 13-17). Finally, we obtain a sparse graph which is globally rigid.

To illustrate the property of global rigidity of this sparse graph, we introduce the following four Lemmas. Meanwhile, formation shapes in different dimensions must satisfy distinct conditions to ensure global rigidity.

- 1) **Linear Formations:** As the vertices in v^{bas} are inherently non-coincident in linear formations. According to **Lemma 4**, \mathcal{G}^{bas} is globally rigid. Since \mathcal{G}_i^{rnn} satisfies the conditions in **Lemma 1 & 2**, \mathcal{G}_i^{rnn} is also globally rigid. Then, the sparse graph $\mathcal{G}^{spr} = \mathcal{G}_1^{rnn} \cup \dots, \mathcal{G}_i^{rnn} \cup \dots, \cup \mathcal{G}_m^{rnn}$, i.e., \mathcal{G}^{spr} is formed after all vertices in v^{rnn} are connected to all vertices in v^{bas} . As the constructed sparse graph satisfies the conditions in **Lemma 3**, \mathcal{G}^{spr} is a globally rigid graph.
- 2) **Planar Formations:** The vertices in v^{bas} are non-collinear in planar formations, then \mathcal{G}^{bas} is globally rigid. Because for collinear vertices, the resulting vertices in v^{bas} are coplanar regardless of how another vertex is added to connect other vertices. According to **Lemma 2**, \mathcal{G}^{spr} is not globally rigid in 3D as the vertices in v^{bas} are coplanar. Similarly, \mathcal{G}^{spr} is a globally rigid graph based on the above explanation.
- 3) **Solid Formations:** The vertices in v^{bas} are non-collinear in planar formations, then \mathcal{G}^{bas} is globally rigid as v^{bas} satisfies the conditions in **Lemma 2**. According to the above explanation, \mathcal{G}^{spr} is a globally rigid graph. ■

Although for **Linear Formations** and **Planar Formations**, two vertices are sufficient to ensure non-collinear of v^{bas} , and three vertices can guarantee non-collinear of v^{bas} . To ensure the formation performance and algorithmic simplicity, we uniformly configure $l|v^{bas}| \geq 4$.

Algorithm 1: Construction of globally rigid graph

```

1 Notation: Number of selected vertices  $m$ , Vertex
   $v^{rnn}$ , Edge  $e^{rnn}$ , Graph  $\mathcal{G}^{rnn}$ 
2 begin
3   if  $m < 4$  then
4      $v^{bas} = v^{cmp}$ 
5      $e^{bas} \subset v^{bas} \times v^{bas}$  (Vertices fully connect)
6      $\mathcal{G}^{spr} = (v^{bas}, e^{bas})$ 
7     return  $\mathcal{G}^{spr}$ 
8   else
9      $v^{bas} = \text{SelectionVbas}(v^{com}, m)$ 
10     $e^{bas} \subset v^{bas} \times v^{bas}$ 
11     $\mathcal{G}^{bas} = (v^{bas}, e^{bas})$ 
12     $v^{rnn} = v^{cmp} - v^{bas}$ 
13    for  $v_i \in v^{rnn}$  do
14       $v_i^{rnn} = v^{bas}.append(v_i)$  (Add  $v_i$  to  $v^{bas}$ )
15       $e_i^{rnn} \subset v_i^{rnn} \times v_i^{rnn}$ 
16       $\mathcal{G}_i^{rnn} = (v_i^{rnn}, e_i^{rnn})$ 
17       $\mathcal{G}^{spr} = \mathcal{G}^{bas} \cup \mathcal{G}_i^{rnn}$ 
18    return  $\mathcal{G}^{spr}$ 

```

Algorithm 2: SelectionVbas

```

1 Notation: Number of selected vertices  $m$ , Vertex
   $v^{rnn}$ , Edge  $e^{rnn}$ , Graph  $\mathcal{G}^{rnn}$ 
2 begin
3   switch Formation shape do
4     case line do
5        $v^{bas} = \text{GRPFVertexSelset}(v^{com}, m)$ 
6     case plane do
7        $v^{bas} = \text{GRPFVertexSelset}(v^{com}, m)$ 
8       if IsCollinear( $v^{bas}$ ) then
9          $v^{rnn} = v^{cmp} - v^{bas}$ 
10         $v_i^{add} = \text{RandomVertexSelset}(v^{rnn}, 1)$ 
11         $v^{bas}.append(v_i^{add})$ 
12     case solid do
13        $v^{bas} = \text{GRPFVertexSelset}(v^{com}, m)$ 
14       if IsCoplanar( $v^{bas}$ ) then
15          $v^{rnn} = v^{cmp} - v^{bas}$ 
16          $v_i^{add} = \text{RandomVertexSelset}(v^{rnn}, 1)$ 
17          $v^{bas}.append(v_i^{add})$ 
18   return  $v^{bas}$ 

```

Definition 3 (*The cone of a graph*) [51]: The cone of a graph \mathcal{G} is obtained from \mathcal{G} by adding a new vertex v and new edges from v to each vertex of \mathcal{G} .

Lemma 1 (*Extension lemma*) [52]. Let \mathcal{G}_1 be a graph obtained from a graph \mathcal{G}_2 by adding a new vertex v with K edges incident to v . If \mathcal{G}_2 is globally rigid in \mathbb{R}^d and $K \geq d + 1$, then \mathcal{G}_1 is globally rigid in \mathbb{R}^d , d is the spatial dimension where the graph is located.

Lemma 2 [53]. A graph is globally rigid if it is formed by starting from a clique of four non-coplanar nodes and repeatedly adding a node connected to at least four non-coplanar existing nodes.

Lemma 3 (*Gluing lemma*) [52]. Let \mathcal{G}_1 and \mathcal{G}_2 be graphs with $|V(\mathcal{G}_1) \cap V(\mathcal{G}_2)| = K$, and let $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$. If \mathcal{G}_1 and \mathcal{G}_2 are globally rigid in \mathbb{R}^d and $K \geq d + 1$, then \mathcal{G} is globally rigid in \mathbb{R}^d . $V(\mathcal{G})$ represents the set of vertices in graph \mathcal{G} , and $|V(\mathcal{G})|$ denotes the number of vertices.

Lemma 4 [51]. A graph \mathcal{G} is globally rigid in \mathbb{R}^d if and only if the cone of \mathcal{G} is globally rigid in \mathbb{R}^{d+1} .

Since the graph is described by the Laplacian matrix, choosing partial edges from complete graphs is equivalent to the submatrix selection. We transform the sparsification mechanism into the submatrix selection, i.e., extracting several columns from the Laplacian matrix. The Laplacian matrix of complete graphs is \mathbf{L}^{cmp} , then \mathbf{L}^{spr} describes the formation constraints in the planning problem that are constructed by submatrix selection. It represents agent i do not consider agent j as a formation constraint when $\mathbf{L}_{i,j}^{spr} = 0$. Fig. 4 shows a graphical interpretation of submatrix selection and graph sparsification.

Through the submatrix selection, a sparse graph with the same vertices as in the complete graphs, which is globally rigid, can be obtained. Considering the vertices as agents in a distributed swarm system, it can be observed that the amount of positional information that needs to be taken into account for each agent in the sparse graph is less than that in the complete graph, thereby reducing the number of formation constraints.

IV. GRPF SPARSE GRAPH CONSTRUCTION METHOD

In this section, we construct a sparse graph that is globally rigid and adequately captures the main features of the original complete graphs through submatrix selection. We compare different submatrix metrics and obtain a closed-form solution for our problem through analysis.

A. GRPF Sparse Graph Construction Problem

According to Section III-B, after determining the connection rate $\varrho_c \in (0, 100\%)$ according to the limited computational resources on board, we round up to obtain the number of deleted columns, which is $\lceil \varrho_c \times N \rceil$. It is evident that the outcome of random submatrix selection is not unique, in order to guarantee formation performance and obtain an optimal submatrix, \mathcal{L}^{spr} should be selected carefully to construct a GRPF sparse graph that can adequately capture the main features of the original complete graphs [35]. As shown in Algorithm 2, we implement the function **GRPFVertexSelset** based on submatrix selection. We formulate the submatrix selection as the following combinatorial optimization problem:

$$P1 : \min_{H^{clm} \subseteq \{1, 2, \dots, N\}, |H^{clm}|=k} \|\mathbf{L}^{cmp} - \mathbf{L}_{[H^{clm}]}^{cmp}\|_2, \quad (7)$$

where H^{clm} is the optimization variable and contains the index subsets of selected column blocks from the Laplacian matrix \mathbf{L}^{cmp} . $\mathbf{L}_{[H^{clm}]}^{cmp} \in \mathbb{R}^{N \times N}$ is the column-selected submatrix, i.e., only the column blocks with the index in H^{clm} have elements that are identical to those in \mathbf{L}^{cmp} , and the rest of the column blocks are $\mathbf{0} \in \mathbb{R}^{N \times 1}$. Then we obtain $\mathbf{L}^{spr} = \mathbf{L}_{[H^{clm}]}^{cmp}$.

Although the combinatorial optimization problem of submatrix selection in P1 can be solved by enumeration, the exponentially growing problem dimension encounters a combinatorial explosion for large-scale formation. The computational complexity of the objective function in P1 is $\mathcal{O}(N^3)$ [54], which is impractical to solve combinatorial optimization problems. Inspired by [35], the matrix-revealing metric is adopted to replace the time-cost objective function, allowing the optimization problem in P1 to be converted into:

$$P2 : \max_{H^{clm} \subseteq \{1, 2, \dots, N\}, |H^{clm}|=k} \mathcal{R}(\mathbf{L}_{[H^{clm}]}^{cmp}), \quad (8)$$

where $\mathcal{R}(\cdot)$ is the matrix-revealing metric. There exist several commonly used metrics which are listed in Table I.

The objectives of P1 and P2 are to maintain the structural information of the complete original graph, as verified through simulations in Section VII.

TABLE I
CANDIDATE MATRIX-REVEALING METRICS

<i>Min-Cond</i> [35]	Minimize condition $\lambda_{\max}(\mathbf{L})/\lambda_{\min}(\mathbf{L})$
<i>Max-logDet</i> [36]	Maximize Log of determinant $\log(\det(\mathbf{L}))$
<i>Max-MinEigenValue</i> [55]	Maximize minimal eigenvalue $\lambda_{\min}(\mathbf{L})$
<i>Max-Trace</i> [56]	Maximize trace $Tr(\mathbf{L}) = \sum_1^N L_{ii}$

λ_{\min} & λ_{\max} are minimal & maximal eigenvalues of square matrix.

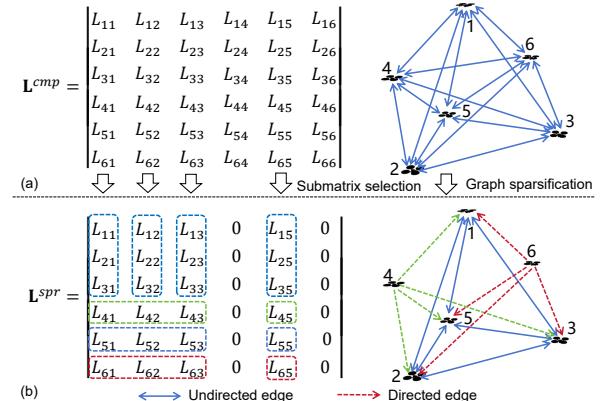
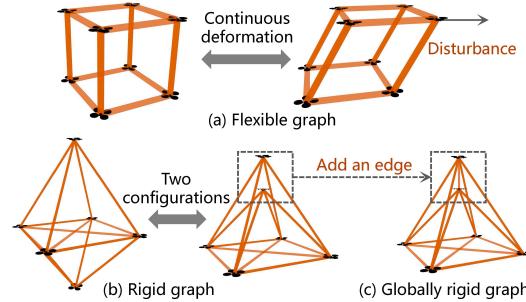


Fig. 4. Illustration of submatrix selection. (a) Laplacian matrix of complete graphs with six vertices. (b) Laplacian matrix of sparse graphs obtained by submatrix selection. The elements in the dashed box correspond to the connected edges in the sparse graph, represented by the same color. Four columns of the matrix are selected to form the submatrix, resulting in that vertices 1, 2, 3, & 5 being selected as the base set v^{bas} . Vertices 1, 2, 3, & 5 connect with each other using undirected edges to establish a complete graph, and vertices 4 & 6 connect with v^{bas} via directed edges.

B. Solution of Sparse Graph Construction Problem

To explore which matrix-revealing metric is most suitable for assisting submatrix selection, we perform a simulation to assess four candidates listed in Table I and simulate three formation configurations (named cube, triangular prism and octahedron) in a cluttered environment of 60m × 20m size, as illustrated in Figs. 5(a)&5(b). To avoid unfair comparisons caused by coupling effects of control, local perception, and other modules, we plan a global trajectory of 40 meters for each agent offline. The numbers of drones 24, 36, and 48 are tested for each configuration. The simulation platform is a personal computer with an Intel Core i7 8700K CPU running at 3.2 GHz and with 32-GB RAM at 3200 MHz. We implement our proposed method in C++.

The sparse graphs obtained using different matrix-revealing metrics are employed in formation planning. We

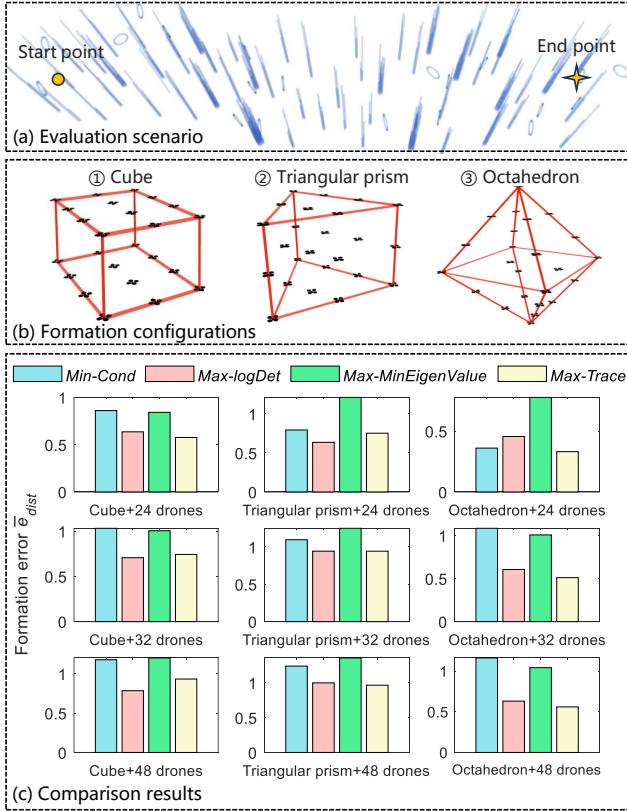


Fig. 5. Comparative simulation of four candidate matrix-revealing metrics. (a) We simulate drones flying in formation from the left side to the right side of a cluttered map with a velocity limit of 2m/s. The map is generated randomly. The global formation trajectory is planned from the start point to end point. (b) Three formation configurations are evaluated. (c) Comparison results of formation error \bar{e}_{dist} along the global trajectory are presented.

refer to the measurement method in [1] to acquire the average formation error \bar{e}_{dist} as a quantitative indicator.

$$\bar{e}_{dist} = \frac{1}{L_{\max}^{fma} L^{trj}} \int_{\mathcal{L}} \min_{\mathbf{R}, \mathbf{t}, s} \sum_{i=1}^N \|\mathbf{p}_i^{des} - (s\mathbf{R}\mathbf{p}_i^{atu} + \mathbf{t})\|_2 dl, \quad (9)$$

where L_{\max}^{fma} and L^{trj} are the maximum diagonal length in the formation and the length of swarm trajectory \mathcal{L} , respectively. l is the length of the trajectory. The combination of rotation $\mathbf{R} \in SO(3)$, translation $\mathbf{t} \in \mathbb{R}^3$, and scale expansion $s \in \mathbb{R}_+$ represents the similarity transformation that aligns the actual flight formation \mathcal{A}^{atu} with the desired one \mathcal{A}^{des} to normalize the formation configuration. Our evaluation criteria differs from [1] in the calculation of the measurement error by replacing the initial formation scale in [1] with a parameter L_{\max}^{fma} , which carries explicit physical significance and denotes the physical scale of the formation. By dividing the formation error by L_{\max}^{fma} , it ensures that formations of different physical sizes exhibit consistent error magnitudes under identical performance. \mathbf{p}_i^{des} and \mathbf{p}_i^{atu} denote the position of the i th drone in \mathcal{A}^{atu} and \mathcal{A}^{des} . Since the influence of rotation, translation, scaling, and trajectory length is eliminated by normalization, \bar{e}_{dist} can fairly measure the degree of distortion between \mathcal{A}^{atu} and \mathcal{A}^{des} along the global formation trajectories.

The comparison results among different matrix-revealing

metrics are presented in Fig. 5(c). The \bar{e}_{dist} values for *Max-Trace* and *Max-logDet* are comparable and clearly outperform the other two metrics. The computational complexity of *Max-Trace* is only $\mathcal{O}(N)$, while for *Max-logDet* the cost is $\mathcal{O}(N^3)$ [35]. Thus, *Max-Trace* is selected as the scoring metric to guide good submatrix selection.

C. Closed-Form Solution of GRPF Sparse Construction

We further explore the geometric feature of *Max-Trace* reflected in the sparsification. Assume that there are k vertices in the base set, and H^{clm} is a solution of P2. We can obtain a vector $\mathbf{v}_H = [v_H[0], \dots, v_H[m], \dots, v_H[N]]$ of dimensional N containing only 0 and 1 elements based on H^{clm} , where

$$v_H[m] = \begin{cases} 1, & \text{if } m \in H^{clm}, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

According to the formula (2)-(4) in Section III-A,

$$tr\{\mathbf{L}_{[H^{clm}]}^{cmp}\} = \sum_{i=0}^N \sum_{j=0}^N A_{i,j}^{cmp}, j \notin H^{clm}. \quad (11)$$

where $tr\{\cdot\}$ denotes the trace of a matrix, \mathbf{A}^{cmp} denotes the adjacency matrix corresponding to the complete graph. Since \mathbf{L}^{cmp} is the symmetric matrix, and the values of the elements in the adjacency matrix are calculated by $\|\cdot\|_2$, $A_{i,j}^{cmp} = A_{j,i}^{cmp}$. It can be inferred that $\sum_{i=0}^N \sum_{j=0}^N A_{i,j}^{cmp}, j \notin H^{clm} \Leftrightarrow \sum_{i=0}^N \sum_{j=0}^N A_{i,j}^{cmp}, i \notin H^{clm}$, where $\sum_{j=0}^N A_{i,j}^{cmp} = L_{i,i}^{cmp} = D_{i,i}^{cmp}$, and \mathbf{D}^{cmp} denotes the degree matrix corresponding to the complete graph. We define the degree vector $\mathbf{Dv} = [D_{0,0}^{cmp}, \dots, D_{i,i}^{cmp}, \dots, D_{N,N}^{cmp}]$. According to the definition of \mathbf{v}_H , we obtain

$$\sum_{i=0}^N \sum_{j=0}^N A_{i,j}^{cmp}, i \notin H^{clm} \Leftrightarrow \mathbf{v}_H \cdot \mathbf{Dv}^T. \quad (12)$$

Consequently, we obtain:

$$\max_{H^{clm} \subseteq \{1, 2, \dots, N\}, |H^{clm}|=k} \mathcal{R}(\mathbf{L}_{[H^{clm}]}^{cmp}) \Leftrightarrow \max(\mathbf{v}_H \cdot \mathbf{Dv}^T). \quad (13)$$

\mathbf{v}_H is equivalent to a selection vector, selecting k elements from the degree vector to sum up. Getting the optimal solution H^{clm*} of problem P2, equivalent to selecting the largest top k elements from \mathbf{Dv}^T . Then, the problem p2 can be solved without optimization. It is evident that the diagonal element in each column of \mathbf{L}^{cmp} (i.e., the element in \mathbf{Dv}^T) represents the sum of the distances between the corresponding vertex and the other vertices. Considering that *Max-Trace* attempts to select the vertex with the maximum diagonal element in \mathbf{L}^{cmp} , it reflects that the selected vertex is positioned in the outline of the formation geometry.

Thus, the closed-form solution of GRPF sparse construction based on *Max-Trace* can be obtained by

$$H^{clm} = \{ \dots, S_{\max}[\mathbf{L}^{cmp}, i], \dots, S_{\max}[\mathbf{L}^{cmp}, k], \dots \}, \quad (14)$$

where $S_{max}[\mathbf{L}^{cmp}, k]$ represents the column index of the selected first k largest diagonal elements in matrix \mathbf{L}^{cmp} . In this way, we enable efficient selection of the submatrix with specified connection rates, facilitating the real-time construction of a GRPF sparse graph.

V. ELIMINATION OF ABNORMAL AGENT STATES

Due to the imperfect hardware and software characteristics of the actual system, agent failures are highly likely to occur in large-scale formation systems. This section introduces our method for eliminating abnormal state information during agent trajectory optimization, thereby obtaining the state information of normal agents and utilizing this information for formation coordination. Since we use graphs to represent the similarity of formations, the solution process effectively involves selecting the maximum clique from the graph that represents the actual relative positions, which exhibits minimal deviation from the expected formation. This approach allows us to effectively identify the majority of normal agents.

We utilize the Laplacian matrix to describe the structure of graphs, which conveniently allows us to measure the deviation from the expected formation shape. The normalized Laplacian matrix corresponding to the actual relative positions is denoted as $\hat{\mathbf{L}}_{cur} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ and the expected Laplacian matrix is $\hat{\mathbf{L}}_{des}$. The difference matrix is denoted as $\Delta\hat{\mathbf{L}} = \hat{\mathbf{L}}_{cur} - \hat{\mathbf{L}}_{des}$, where the off-diagonal elements of $\Delta\hat{\mathbf{L}}$ represent the errors corresponding to the lengths of the edges. By setting an appropriate threshold for edge length errors based on the environment and the size of the formation, we can identify edges within $\Delta\hat{\mathbf{L}}$ that exhibit excessively large errors. Subsequently, we disconnect these edges in the complete graph composed of all agents, resulting in a sparse graph. By computing the maximum clique of this sparse graph, we can identify the majority of normal agents. The complete graph corresponding to the maximum clique is denoted as $\mathcal{G}_{mc} = (\mathbf{v}_{mc}, \mathbf{e}_{mc})$, where \mathbf{v}_{mc} and $\mathbf{e}_{mc} \subset \mathbf{v}_{mc} \times \mathbf{v}_{mc}$ represent the set of vertices and edges in the maximum clique, and the associated Laplacian matrix is denoted as \mathbf{L}_{mc} . Thus, we transform the task of removing anomalous agent state information into the problem of computing the maximum clique.

In order to obtain the maximum clique while accommodating limited computation resources in real time in large-scale scenarios, we employ the calculation of the maximum k -core as an approximation of the maximum clique, and the elimination process is initiated at a low frequency which is below 2 Hz. The k -core is the maximal subgraph \mathcal{C}_k of \mathcal{G} in which every vertex within \mathcal{C}_k is adjacent to at least k vertices [57]. [58] observe that the maximum clique resides within the maximum k -core in practical problems and solving for k -core is of linear complexity [12].

VI. FORMATION PLANNING BASED ON SPARSE GRAPH

After eliminating abnormal agents, we perform formation trajectory planning based on the corresponding collaborative

relationships of the sparse graph.

A. Trajectory Representation

We embrace \mathfrak{T}_{MINCO} [59] as the foundation for our trajectory representation. This class defines a polynomial trajectory with minimal control effort, characterized by:

$$\mathfrak{T}_{MINCO} = \{p(t) : [0, T_\Sigma] \rightarrow \mathbb{R}^m | \mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T}), \mathbf{q} \in \mathbb{R}^{m(M-1)}, \mathbf{T} \in \mathbb{R}_{>0}^M\}, \quad (15)$$

where $p(t)$ represents an m -dimensional M -piece polynomial trajectory of degree $N = 2s - 1$ and s denotes the order of the relevant integrator chain. The polynomial coefficient $\mathbf{c} = (\mathbf{c}_1^T, \dots, \mathbf{c}_M^T)^T \in \mathbb{R}^{2Ms \times m}$ is derived from $\mathcal{M}(\mathbf{q}, \mathbf{T})$, where $\mathbf{q} = (\mathbf{q}_1, \dots, \mathbf{q}_{M-1})$ represents the intermediate waypoints and $\mathbf{T} = (T_1, T_2, \dots, T_M)^T$ denotes the time allocated for each piece. $T_\Sigma = \sum_{i=1}^M T_i$ signifies the total time span.

Each m -dimensional M -piece trajectory is defined as follows:

$$p(t) = p_i(t - t_{i-1}) \quad \text{for all } t \in [t_{i-1}, t_i], \quad (16)$$

where the i^{th} piece trajectory is a $N = 5$ degree polynomial:

$$p_i(t) = \mathbf{c}_i^T \beta(t) \quad \text{for all } t \in [0, T_i], \quad (17)$$

with $\mathbf{c}_i \in \mathbb{R}^{(N+1) \times m}$ as the coefficient matrix, $\beta(t) = [1, t, \dots, t^N]^T$ as the natural basis, and $T_i = t_i - t_{i-1}$ as the time allocated for the i^{th} piece. The uniqueness of \mathfrak{T}_{MINCO} is determined by (\mathbf{q}, \mathbf{T}) . Parameter mapping $\mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T})$ facilitates the conversion of trajectory representations (\mathbf{c}, \mathbf{T}) to (\mathbf{q}, \mathbf{T}) , allowing any second-order continuous cost function $J(\mathbf{c}, \mathbf{T})$ to be represented by $H(\mathbf{q}, \mathbf{T}) = J(\mathcal{M}(\mathbf{q}, \mathbf{T}), \mathbf{T})$.

To address time-integral constraints such as collision avoidance, dynamical feasibility, and traversability cost, we discretize each piece of the trajectory into κ_i constraint points $\tilde{p}_{i,j} = p_i((j/\kappa_i) \cdot T_i)$ for $j = 0, 1, \dots, \kappa_i - 1$.

B. Problem Formulation

We adopt a framework of distributed trajectory optimization. Formation planning is formulated as a constrained optimization problem:

$$\min_{\mathbf{q}, \mathbf{T}} \int_{t_0}^{t_M} \|p^{(s)}(t)\|^2 dt + \rho \cdot T_\Sigma, \quad (18)$$

$$\text{s.t. } p(t) = \mathcal{M}_{\mathbf{q}, \mathbf{T}} \quad \forall t \in [t_0, t_M], \quad (19)$$

$$\mathbf{p}^{[s-1]}(0) = \bar{\mathbf{p}}_0, \quad (20)$$

$$\mathbf{p}^{[s-1]}(t_M) = \bar{\mathbf{p}}_f, \quad (21)$$

$$\mathcal{H}(p(t), \dots, p^{(s)}(t)) \leq 0 \quad \forall t \in [t_0, t_M]. \quad (22)$$

The robot state $p(t)$ is parameterized by the optimization variables $\{\mathbf{q}, \mathbf{T}\}$ and ρ is the time regularization parameter. The continuous-time constraint \mathcal{H} includes similarity in group formation, dynamic feasibility, obstacle avoidance, and reciprocal avoidance of swarms. $\mathbf{p}^{[s-1]}(t) = (p(t)^T, \dot{p}(t)^T, \dots, p^{[s-1]}(t)^T)^T \in \mathbb{R}^{ms}$ represents the higher-order derivative of a chain dynamical system with an s -integrator. $\bar{\mathbf{p}}_0$ represents the initial state and $\bar{\mathbf{p}}_f$ represents the terminal state.

We use optimization variables of MINCO (15) to eliminate the equality constraints (19)-(21) to solve continuous constraint optimization problems in real time. And the penalty function method [60] is used to handle inequality constraints (22), reader can refer to [59] for specific methods of constraint elimination and optimality assurance. Then we transform the continuous constrained optimization problem into a discrete unconstrained optimization problem:

$$P3 : \min [\mathcal{J}_f, \mathcal{J}_o] \cdot \mu, \quad (23)$$

where $\mu = \{\mu_f, \mu_e, \mu_t, \mu_c, \mu_s, \mu_d\}$ denotes the weight vector. We refer to the proportions of weight in the same distributed unconstrained trajectory optimization framework, where the obstacle avoidance weight was set higher for safety. The specific values of these parameters in this paper are detailed in Table II. \mathcal{J}_o contains the cost function of the control effort, flight time, collision avoidance, swarm reciprocal avoidance, dynamical feasibility, and modeling using the same method as [1], the readers can refer to [1] for more details. $\mathcal{J}_f = f(\mathcal{F}_f)$ defines the cost function of similarity formation, and $f(\cdot)$ is the differentiable metric to quantify the similarity distance $\mathcal{F}_f = \|\mathbf{L}_{mc}^{sqr} - (\mathbf{L}_{mc}^{sqr})_{des}\|_F^2$ between current and desired formations. \mathbf{L}_{mc}^{sqr} is the Laplacian matrix that represents the sparse graph \mathcal{G}_{mc}^{spr} , and $(\mathbf{L}_{mc}^{sqr})_{des}$ denotes the matrix describing the desired formation configuration in the maximum clique. $\|\cdot\|_F$ is Frobenius norm. Although we employ the same differentiable function to compute similarity distances like [1], the Laplacian matrix utilized as the metric is post-processed through outlier rejection and sparse graphs.

While such an unconstrained optimization problem cannot theoretically guarantee full constraint satisfaction, in the specific trajectory planning process, to ensure obstacle avoidance safety, we employ the A-star algorithm to generate an initial path as the initial value for the trajectory optimization problem. This path serves as a high-quality initial value that inherently incorporates obstacle avoidance. Additionally, we give greater weight to obstacle avoidance to prioritize safety. Moreover, to guarantee dynamic feasibility along the trajectory, we enforce conservative upper bounds on the dynamics of the agents in the optimization formulation, which are lower than the actual executable dynamic limits. Furthermore, our 10Hz local replanning system discards non-converged solutions immediately via rigorous verification, preventing constraint avoidance.

TABLE II

WEIGHT PARAMETERS FOR TRAJECTORY OPTIMIZATION PROBLEMS

Weight	Symbol	Value
Weight for the formation similarity	μ_f	200.0
Weight for the control effort	μ_e	100.0
Weight for the flight time	μ_t	80.0
Weight for the collision avoidance	μ_c	5000.0
Weight for the swarm reciprocal avoidance	μ_f	5000.0
Weight for the dynamical feasibility	μ_d	100.0

C. Complexity Analysis

We analyze the computational complexity of P3 and derive the gradient of \mathcal{F}_f , since the cost of swarm formation similarity dominates the computationally expensive element. According to the chain rule, the gradient of \mathcal{F}_f with respect to the position of drones $p_i(t)$ can be obtained by

$$\frac{\partial \mathcal{F}_f}{\partial p_i(t)} = \frac{\partial \mathcal{F}_f}{\partial \mathbf{w}_i^T} \frac{\partial \mathbf{w}_i^T}{\partial p_i(t)}, \quad (24)$$

where

$$\frac{\partial \mathcal{F}_f}{\partial \mathbf{w}_i^T} = \left[\frac{\partial \mathcal{F}_f}{\partial \mathbf{w}_{i1}}, \dots, \frac{\partial \mathcal{F}_f}{\partial \mathbf{w}_{ij}}, \dots, \frac{\partial \mathcal{F}_f}{\partial \mathbf{w}_{in}} \right], \quad j \in \mathcal{N}_i, \quad (25)$$

$$\frac{\partial \mathbf{w}_i^T}{\partial p_i(t)} = \left[\frac{\partial \mathbf{w}_{i1}}{\partial p_i(t)}, \dots, \frac{\partial \mathbf{w}_{ij}}{\partial p_i(t)}, \dots, \frac{\partial \mathbf{w}_{in}}{\partial p_i(t)} \right], \quad j \in \mathcal{N}_i. \quad (26)$$

\mathcal{N}_i is the set of adjacent vertex in the drone i , and \mathbf{w}_i^T is a weight vector composed of edge weights in \mathcal{N}_i . Suppose the worst-case scenario, where all agents are within the maximum clique, according to (25) & (26), the computational complexity of $\partial \mathcal{F}_f / \partial \mathbf{w}_i^T$ and $\partial \mathbf{w}_i^T / \partial p_i(t)$ is $\mathcal{O}(N)$ when using complete graphs, i.e., the number of drones in \mathcal{N}_i is $N - 1$. Thus, $\partial \mathcal{F}_f / \partial p_i(t)$ consumes $\mathcal{O}(N^2)$. For sparse-graph-based formation planning, the computation cost is reduced to $\mathcal{O}((\varrho_c N)^2)$ with the connection rate $\varrho_c \in (0, 100\%)$. If ϱ_c is set to 30%, the complexity is almost one order of magnitude lower than that of complete graphs.

VII. BENCHMARKS AND EXPERIMENTS

Extensive simulations based on a global and local planning framework are carried out, and we also implement physical experiments to validate the performance of sparse-graph-based formation planning methods.

A. Simulation of Efficient Global Planning

We explore the formation error and the efficiency of the computation with respect to various connection rates of the graphs ϱ_c . In order to primarily analyze the performance of planning, we conduct simulations in the global planning framework, i.e., generating a trajectory connecting the starting and ending point without the need of local replanning. The simulation environments and hardware are set as the same in Section IV-B. In addition, we benchmark with other sparse graph construction methods.

1) *Study of Planning Efficiency and Performance*: This subsection focuses on analyzing the computation time used for planning and formation performance with different connection rates of sparse graphs and varying numbers of drones. Fig. 6 depicts the computation time and formation error of different connection rates. We observe that the computation time reduces considerably as ϱ_c decreases in various test scenarios. For 72 drones, the computation efficiency under $\varrho_c = 30\%$ is over 10× faster than the counterpart with $\varrho_c = 100\%$. Although the method suffers from a performance drop as ϱ_c decreases, \bar{e}_{dist} with $\varrho_c = 30\%$ only increases by approximately 30% compared to $\varrho_c = 100\%$. As a result, real-time formation planning with 72 drones can be

achieved when $\varrho_c = 30\%$, and involves an acceptable and adjustable sacrifice in formation error. Formation planning in complete graphs ($\varrho_c = 100\%$) takes seconds and becomes computationally prohibitive for real-time requirements.

2) Benchmark on Formation Keeping: To demonstrate the superiority of our sparse graph construction method, three sparse graph construction methods including random sparse graph (*Random*), nearest neighbor sparse graph (*Nearest*) [4], and our method without optimization on the base set v^{bas} (*Ours w/o opt*) are compared with the proposed method in terms of formation error and formation recovery. The details of implementation of these comparative methods are listed in Table III. For fair comparison, all the constructed graphs are integrated into the same planning problem in Section V. The test scenario involving 48 drones is established to require maintaining an octahedron formation. The connection rate ϱ_c is set as 30%, thus the number of edges connected in each drone $l|v^{bas}|$ is set to 15. Each method is run 80 times with randomly generated maps.

From the comparison results in Fig. 7(a), the average formation error \bar{e}_{dist} and its dispersion of *Nearest* and *Random* are large. *Ours w/o opt* is better and benefits the global rigidity of the constructed sparse graph, but the randomly selected base set makes the formation performance unstable. Our method outraces all the methods in Table III and achieves lower \bar{e}_{dist} , with much tighter dispersion. In addition, \bar{e}_{dist} by ours is very close to that of *Complete*, which can be attributed in part to the better structural information captured by *Ours*. In Fig. 7(b), it can be found that the vertices in the base set describe the skeleton of the graph more holistically than *Ours w/o opt*.

3) Benchmark on Formation Recovery: A swarm with 48 drones is required to recover from a scattered state to octahedron formation without obstacle avoidance, and the connection rate is set to 30%. From Fig. 9(a), formation planning based on a sparse graph highlights an order of magnitude's speedup, w.r.t. *complete*. The formation recovery performance is qualitatively visible by plotting a subset of formation states in Fig. 9(b). We identify that ours can form the desired octahedron shape faster against the first three methods in Table III, and the time required for recovery in Fig. 9(c) is very close to *complete*. In summary, a GRPF sparse graph can facilitate the planing to obtain satisfactory formation results and improve the computation efficiency significantly.

TABLE III
COMPARATIVE METHODS IN BENCHMARK

<i>Random</i>	Each drone randomly selects $l v^{bas} $ drones to connect
<i>Nearest</i>	Each drone selects $l v^{bas} $ nearest drones to connect
<i>Ours w/o opt</i>	Base set v^{bas} is selected randomly without optimization
<i>Complete</i>	Each drone connects with all other drones
<i>Ours</i>	Establishing connection based on GRPF graph

$l|v^{bas}|$ is the number of connected edges of each drone.

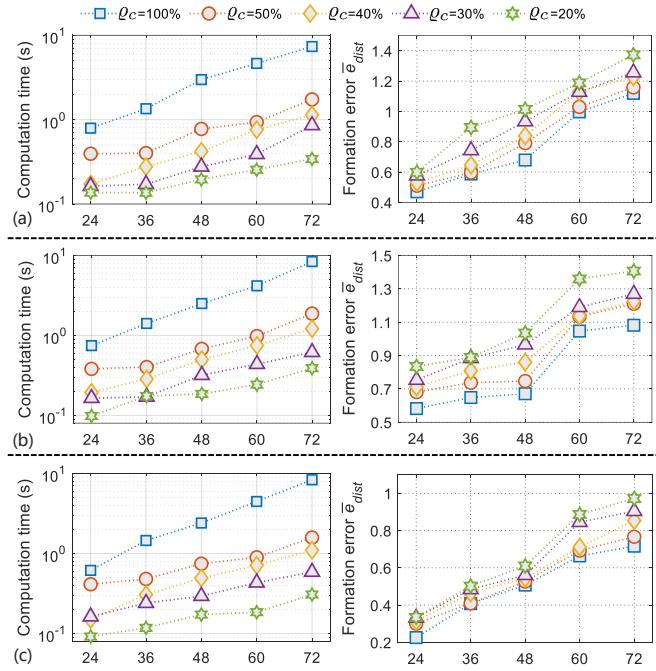


Fig. 6. Comparison results of planning efficiency and formation error. (a)-(c) represent the results on different formation shapes (cube, triangular prism, and octahedron). The connection rates of graphs ϱ_c ranging from 20% to 50% are tested. The number of drones is set as 24, 36, 48, 60, and 72. The first and second columns of the figure represent the computing time and formation error, respectively. The data in the figure are the average value for 20 runs, with the maps randomly regenerated for each test.

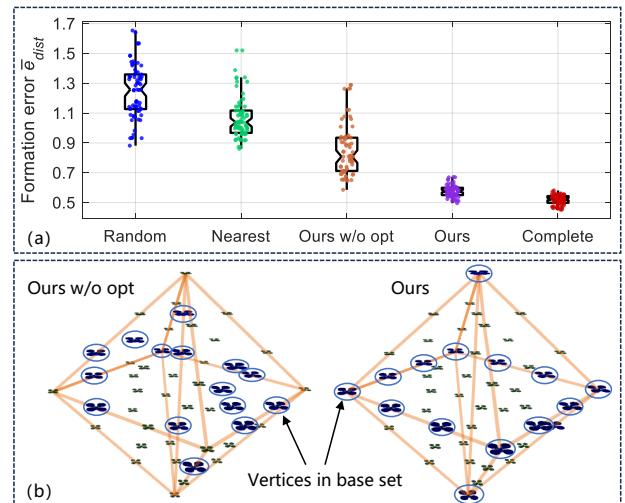


Fig. 7. Comparative simulation on formation error. (a) Boxplot of formation error by different methods. (b) The visualization of base sets selected by *ours w/o opt* and *ours*, respectively.

B. Simulation of Efficient Local Planning

In this subsection, we integrate our sparse graph construction into the distributed local planning framework. We compare with the cutting-edge formation planning method, including Quan's method [1] and the virtual rigid body (VRB) based method [19], to fully demonstrate that our method can achieve a better efficiency-performance trade-off. We configure the connection rate $\varrho_c = 30\%$ and test formation planning with 10 to 100 drones that maintain

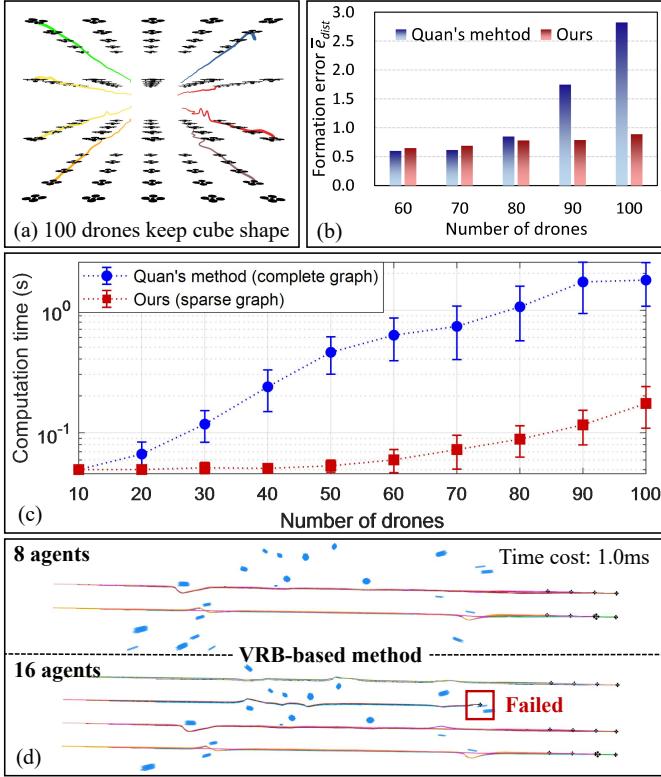


Fig. 8. Comparative simulation with Quan’s method [1] and the VRB-based method in local planning framework. (a) Cube formation with 100 drones. (b) Formation error. (c) Formation planning runtime. (d) Local planning with VRB-based method.

TABLE IV

ABLATION EXPERIMENT OF ELIMINATING OUTLIERS

Number of agents	5	10	15	20
$\bar{e}_{dist}(w)$	0.072	0.066	0.159	0.216
$\bar{e}_{dist}(w/o)$	0.348	0.294	0.371	0.446

$\bar{e}_{dist}(w)$: Formation error of planning with eliminating outliers;
 $\bar{e}_{dist}(w/o)$: Formation error of planning without eliminating outlier.

the shape of a cube in obstacles-rich environments. As the number of drones exceeds 80, the perception and mapping modules nearly exhaust the computational resources in the personal computer used in Section IV-B. To perform simulations for a large-scale swarm with more than 80 drones, we use a high performance workstation which equipped multiple AMD EPYC 7B13 CPU with 256-GB RAM. The formation error is shown in Fig. 8(b). For comparison of computation time, we record the longest time in the local planning process of each drone. The average and variance of the time between all drones are illustrated in Fig. 8(c).

It is evident that as the number of drones increases, the computation time of Quan’s method grows significantly compared to *Ours*. The computation time of *Ours* maintains a low level consistently if the number of drones is less than 50, and the time remains below 100 ms, even when the number reaches 80. In comparison, when the number of drones reaches 80, the trajectory optimization time of Quan’s method costs more than 1 s. Due to the substantial increase in computation time of Quan’s method, the trajectories shared

with other drones become outdated, leading to a larger formation error. As shown in Fig. 8(b), the formation error of *Ours* becomes lower than that of Quan’s method with 80 drones. For VRB-based method, although this category of formation control approaches is computationally inexpensive, it is highly prone to deadlocks in obstacle environments. As illustrated in Fig. 8(d), a formation with 8 agents can navigate through sparse obstacle environments, but when the number increases to 16 agents, the formation fails. Through comparison, we verify that our method maintains a better formation performance for large-scale swarms.

Finally, in order to demonstrate the high-performance and efficient formation trajectory planning effectiveness, we present a large-scale formation planning of 100 agents in an obstacle-rich area that maintains the shape of a jet aircraft with the same hardware in Section IV-B, it takes 0.12 seconds for each local planning, the result is shown in Fig. 1.

C. Ablation study of Globally Rigid Graph Construction

To validate the key part of the Algorithm 1 in Section IV-B (i.e., vertices in v^{bas} should be non-coplanar in 3D), the ablation study is conducted. As shown in Fig. 10(a), we select coplanar vertices to form v^{bas} intentionally. According to the graph rigidity theory in Section III-A, the constructed sparse graph with coplanar v^{bas} may be just a rigid graph instead of a globally rigid graph in 3D, which cannot uniquely determine a geometric shape. Since different shapes correspond to different local minima and these local minima are close to each other, formation planning oscillates among multiple local minima and cannot converge to a feasible solution, resulting in the collapse of formation. Additionally, we simply change the position of a vertex to establish a non-coplanar v^{bas} in Fig. 10(b). It can be observed that the graph with non-coplanar v^{bas} can maintain stable formation for three configurations.

D. Ablation study of Abnormal Agents Elimination

We obtain the maximum clique containing normal agents by computing the maximum k -core, thus eliminating abnormal agents and reducing the impact on the majority agents in the overall formation. By comparing the elimination of abnormal states through ablation experiments, we demonstrate the effectiveness of our algorithm. As shown in Fig. 12, an agent is set far from the desired formation position. Our method can automatically exclude outliers. When the outlier moves closer to the desired position, our approach can smoothly reintegrate it into the formation. The results indicate that the trajectories of most normal agents remain smooth and the overall formation shape is well maintained. In contrast, the method that considers every agent at all times leads to convoluted trajectories for all agents and considerably affects the integrity of the overall formation.

As shown in Table IV, we further compared the formation error with different numbers of agents flying in an obstacle-rich environment, each formation comprising an agent in an abnormal state. As the number of agents increases, the \bar{e}_{dist} tends to increase for both methods. However, when the

TABLE V
COMPARATIVE SIMULATION ON FORMATION RECOVERY

Number of agents	24		36		48		60	
	$t^{rcy}(s)$	$t^{com}(s)$	$t^{rcy}(s)$	$t^{com}(s)$	$t^{rcy}(s)$	$t^{com}(s)$	$t^{rcy}(s)$	$t^{com}(s)$
Random	N/A	0.172	N/A	0.244	N/A	0.289	N/A	0.443
Nearest	N/A	0.168	N/A	0.236	N/A	0.256	N/A	0.430
Ours w/o opt	6.52	0.171	7.88	0.238	11.00	0.288	12.08	0.428
Complete	2.22	0.684	2.64	1.652	4.54	2.982	6.43	4.561
Ours	2.96	0.163	3.42	0.240	5.52	0.276	6.88	0.431

t^{com} : Time used for planning (s); t^{rcy} : Time used for recovery (s); N/A: Formation planning fail, i.e., cannot form desired configuration.

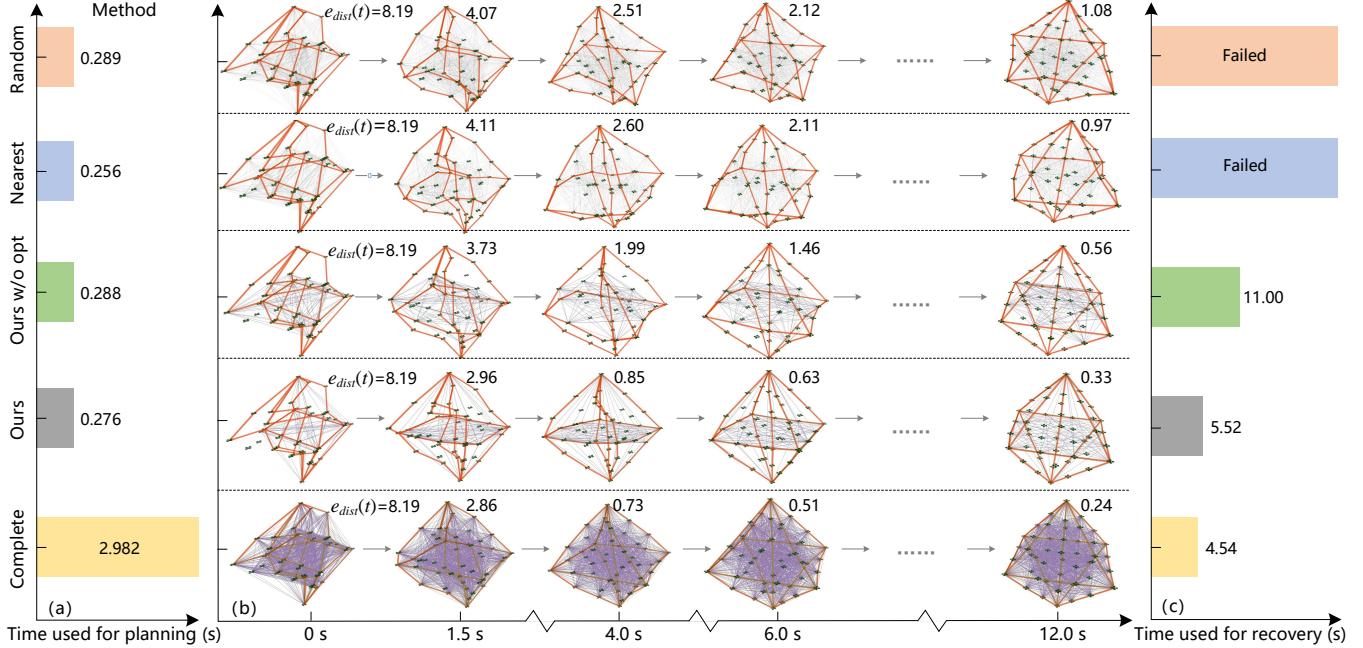


Fig. 9. Comparative simulation on formation recovery. Several formation states are provided at different time. The thick orange line represents the overall skeleton of the geometric shape. The thin grey line is the edge connecting different drones. $e_{dist}(t) = (\min_{\mathbf{R}, \mathbf{t}, s} \sum_{i=1}^N \|\mathbf{p}_i^{des} - (\mathbf{s}\mathbf{R}\mathbf{p}_i^{att} + \mathbf{t})\|_2) / L_{max}^{fma}$ is the instantaneous formation error at time t . If $e_{dist}(t) \leq 0.65$, it is consider that the formation is converged to the desired shape.

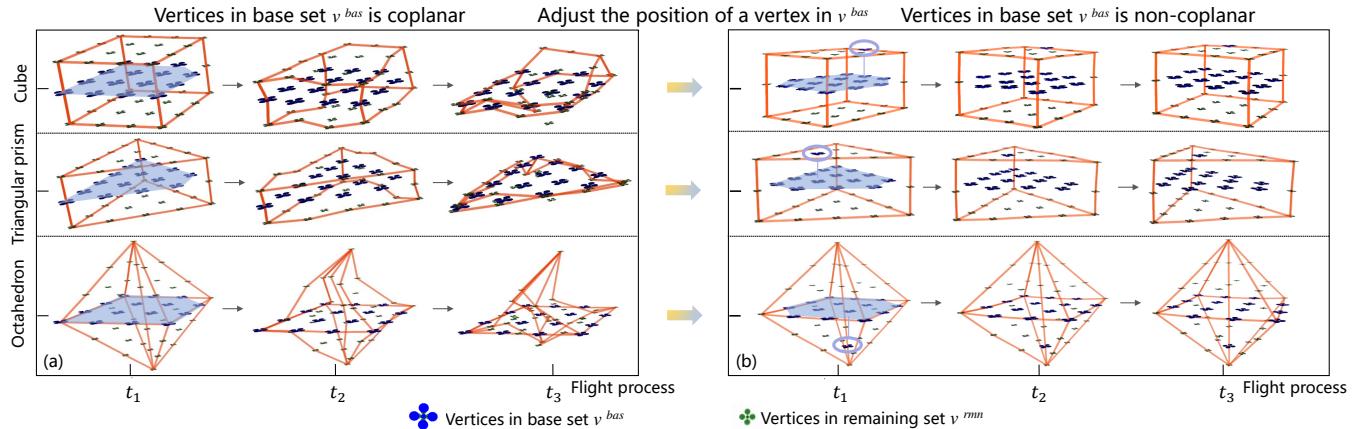


Fig. 10. Formation planning results in the ablation study. Three formation configurations are tested, and the number of drones and connection rate are set as 48 and 30%, respectively. (a) The formation with coplanar vertices in base set v^{bas} . The shaded area represents the vertices in v^{bas} . (b) The formation with non-coplanar vertices in v^{bas} . The vertices in the shaded area and circle represent v^{bas} .

number of agents is less than 20, the method with outlier rejection consistently exhibits a \bar{e}_{dist} that is less than 50% of the value observed in the method without outlier rejection. Specifically, when the number of agents is less than 10, \bar{e}_{dist}

is reduced by as much as 78% in the method with outlier rejection. By comparing the formation error of the swarm without the outlier agent, it can be found that the method of outlier rejection has a significant advantage in ensuring the

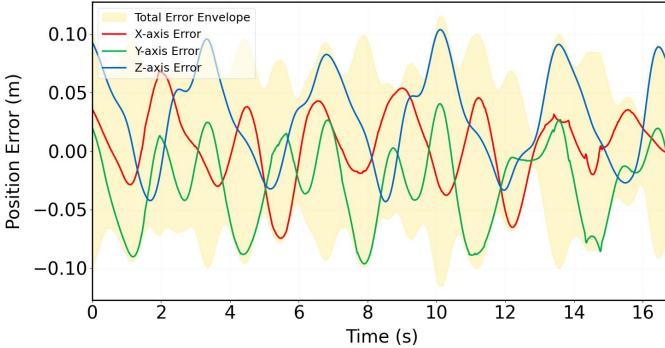


Fig. 11. Three-axis position tracking errors and total error of the SE(3) controller in our drone system.

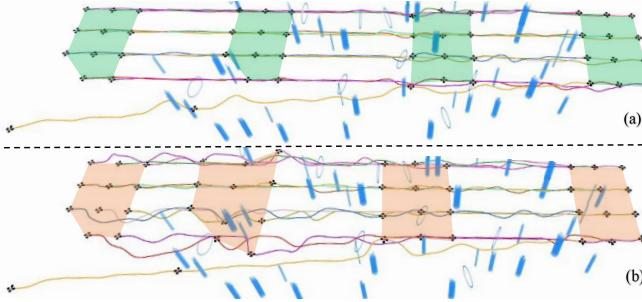


Fig. 12. Ablation experiment of eliminating outliers in local planning framework. (a) with eliminating outliers. (b) w/o eliminating outliers.

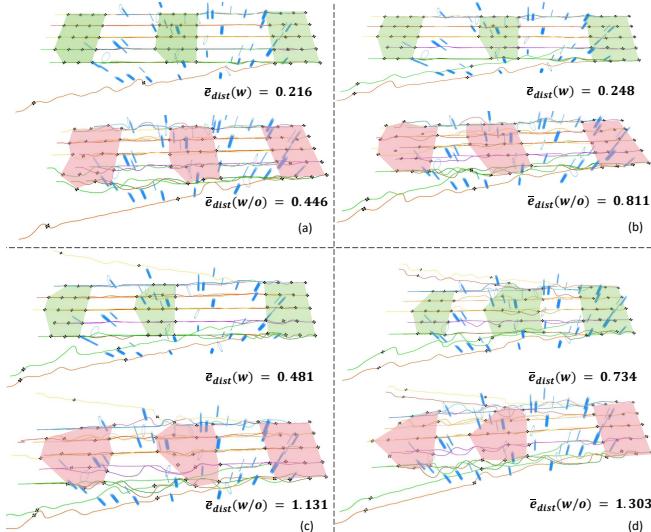


Fig. 13. Ablation experiment of eliminating outliers with different proportion of abnormal agents. Formations in green boxes with outlier rejection show lower normal-agent trajectory disturbance by anomalies, contrasting obviously with distorted paths of formations in red boxes. Due to lacking outlier rejection, The formation maintenance is degraded in red boxes. (a)-(d) show 1, 2, 3, and 4 outliers in formation, respectively.

overall performance of the remaining formation.

Furthermore, we investigate formation planning under different anomalous outlier levels while maintaining identical formation scales. As shown in Fig. 13, compared to methods without outlier rejection, the approach with outlier rejection ensures stable formation coordination among normal agents with smoother overall trajectories. The \bar{e}_{dist} of normal agents are all below 60% of those without outlier rejection, and the minimum \bar{e}_{dist} reaches 30%. However, as the proportion of anomalous individuals increases, as shown in Fig. 13(d), our

method inevitably suffers from trajectory distortion caused by excessive outliers. Even under such conditions, \bar{e}_{dist} remains approximately 56% of the value observed in methods without outlier rejection. Our method can effectively address the issue of abnormal agents.

E. Real World Experiments

We integrate the sparse-graph-based trajectory planning method into the distributed physical aerial swarm system to maintain formation, and the architecture of the physical swarm system architecture is shown in Fig. 14(a). Each drone has independent modules for control, perception, and trajectory planning. The corresponding hardware includes the KAKUTE H7 Mini flight controller for control, the Intel RealSense D435 stereo camera for perception, and the NOKOV Motion Capture System¹ for estimation. The mapping, estimation, and formation planning is run with an onboard computer Xavier NX in real-time.

As shown in Fig. 14(e), drones 0 to 3 correspond to the base vertices v^{bas} in the sparse graph, while drone 4 corresponds to the remaining vertices v^{rem} . The solid blue lines indicate bidirectional connections, whereas the dashed lines represent unidirectional connections. According to *Proposition 1*, this sparse graph is globally rigid. The maximum speed of drones is 1m/s. Each drone relies on local sense for obstacle avoidance without prior mapping. We adopt a classical geometric tracking control method on SE(3) [61] to track the desired trajectory and perform the experiment on an actual drone. As shown in Fig. 11, the tracking errors along the three coordinate axes and the total tracking error are maintained below 0.1m, with negligible degradation observed in formation performance. The experimental results demonstrate that our method can guide the swarm in maintaining formation in obstacle-rich environments. Throughout the entire flight process, the $e_{dist}(t)$ does not exceed 0.3. As shown in Fig. 14(d), the formation experiences a significant increase in $e_{dist}(t)$ due to the interference of obstacles. However, the formation can rapidly restore its desired configuration once it exits the dense obstacle environment.

F. Discussion

The proposed GRPF sparse graph effectively reduces the computation time for trajectory optimization, while demonstrating remarkable formation maintenance and recovery capabilities in complex environments. However, this work does not investigate formation coordination performance under time-varying topological conditions and the connection rate of the sparse graph relies on empirical selection. Considering the swarm reciprocal avoidance, agents still need to receive the trajectories of all other drones. However, polynomial trajectories can be compactly represented by a small number of parameters, which requires a small communication bandwidth and is less than 10 bps. For larger formation planning, communication complexity can be reduced through swarm

¹<https://www.nokov.com>

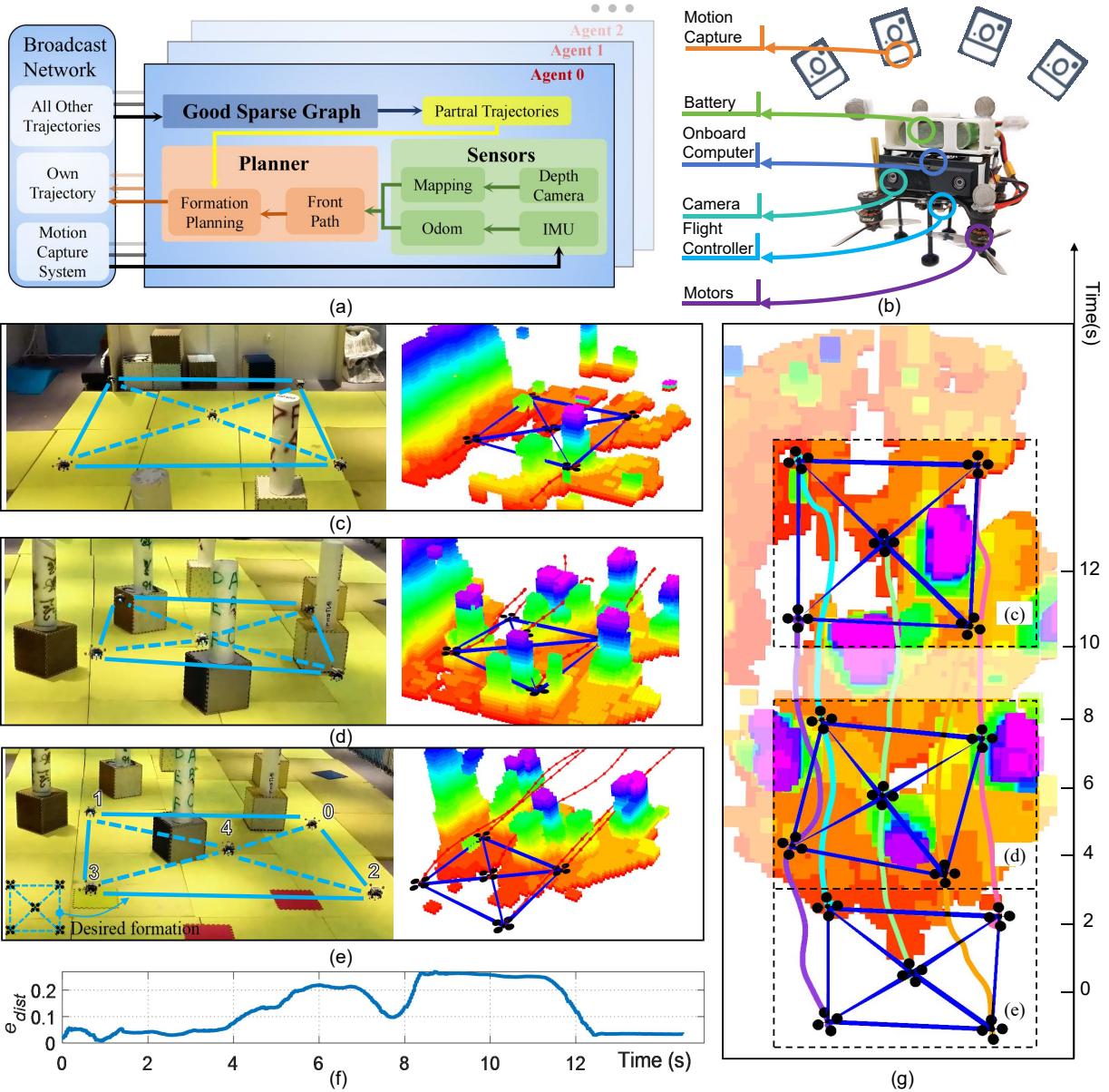


Fig. 14. (a) System architecture of distributed aerial swarm in real world experiments. (b) Quadrotor in real world experiments. (c) Screenshot and Rviz diagram of the final formation. (d) Screenshot and Rviz diagram of the formation corresponding to the 5 second. (e) Screenshot and Rviz diagram of the initial formation. (f) Formation error e_{dist} in the flight process. (g) Rviz diagram of a distributed swarm system of five drones in a square, which uses the sparse graph to model formation constraint, is traversing an unknown obstacle area.

grouping [62]. Furthermore, although anomaly rejection mechanisms achieve simultaneous connectivity sparsification and mitigation of anomalous agent impacts, our approach is currently limited to scenarios with a small proportion of abnormal agents. The rejection threshold remains an empirically determined threshold requiring systematic refinement through data-driven threshold adaptation mechanisms. Additionally, we adopt idealized communication assumptions, without considering the communication delay [63], [64] and link failure [25] in practical scenarios.

Notwithstanding these limitations, our methodology outperforms cutting-edge formation planning methodologies [1], [19] in computational efficiency and anomaly resilience, particularly in large-scale swarm configurations.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we eliminate agents in abnormal states by competing the maximum k -core, ensuring the collaboration of the majority of normal agents. Then, we propose a sparse graph construction method for efficient formation planning. The graph sprasification mechanism and submatrix selection are integrated to ensure that the sprasified graph is globally rigid and captures the main structural feature of complete graphs. Benchmark comparisons and simulation indicate that our method can significantly reduce computational time, while the formation performance remains close to that under the complete graph. The physical autonomous distributed aerial swarm system is used to verify the effectiveness of our method.

In the future, we aim to scale the formation system to larger swarms and investigate the impact of communication delays and link failure. Furthermore, we intend to develop learning-based strategies to achieve more resilient formation maintenance with outlier elimination.

REFERENCES

- [1] L. Quan, L. Yin, T. Zhang, M. Wang, R. Wang, S. Zhong, X. Zhou, Y. Cao, C. Xu, and F. Gao, “Robust and efficient trajectory planning for formation flight in dense environments,” *IEEE Transactions on Robotics*, 2023.
- [2] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, “Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm,” in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 8776–8782.
- [3] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable self-assembly in a thousand-robot swarm,” *Science*, vol. 345, no. 6198, pp. 795–799, 2014.
- [4] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, “Region-based shape control for a swarm of robots,” *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.
- [5] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Probabilistic and distributed control of a large-scale swarm of autonomous agents,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1103–1123, 2017.
- [6] G. Sun, R. Zhou, Z. Ma, Y. Li, R. Groß, Z. Chen, and S. Zhao, “Mean-shift exploration in shape assembly of robot swarms,” *Nature Communications*, vol. 14, no. 1, p. 3476, 2023.
- [7] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, “Towards a swarm of agile micro quadrotors,” *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.
- [8] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, “Swarm of micro flying robots in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.
- [9] M. Ballerini, N. Cabibbo, R. Candier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini *et al.*, “Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study,” *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1232–1237, 2008.
- [10] D. W. Sankey, R. F. Storms, R. J. Musters, T. W. Russell, C. K. Hemelrijk, and S. J. Portugal, “Absence of “selfish herd” dynamics in bird flocks under threat,” *Current Biology*, vol. 31, no. 14, pp. 3192–3198, 2021.
- [11] C. Boutsidis, M. W. Mahoney, and P. Drineas, “An improved approximation algorithm for the column subset selection problem,” in *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, 2009, pp. 968–977.
- [12] N. S. Dasari, R. Desh, and M. Zubair, “Park: An efficient algorithm for k-core decomposition on multicore processors,” in *2014 IEEE international conference on big data (Big Data)*, pp. 9–16.
- [13] R. Chai, Y. Guo, Z. Zuo, K. Chen, H.-S. Shin, and A. Tsourdos, “Cooperative motion planning and control for aerial-ground autonomous systems: Methods and applications,” *Progress in Aerospace Sciences*, vol. 146, p. 101005, 2024.
- [14] J. Alonso-Mora, S. Baker, and D. Rus, “Multi-robot formation control and object transport in dynamic environments via constrained optimization,” *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.
- [15] B. Açıkmeşe and D. S. Bayard, “Markov chain approach to probabilistic guidance for swarms of autonomous agents,” *Asian Journal of Control*, vol. 17, no. 4, pp. 1105–1124, 2015.
- [16] F. Trotti, A. Farinelli, and R. Muradore, “A markov decision process approach for decentralized uav formation path planning,” in *2024 European Control Conference (ECC)*. IEEE, pp. 436–441.
- [17] L. Sabattini, C. Secchi, and C. Fantuzzi, “Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation,” *Autonomous Robots*, vol. 30, pp. 385–397, 2011.
- [18] M. A. Lewis and K.-H. Tan, “High precision formation control of mobile robots using virtual structures,” *Autonomous robots*, vol. 4, pp. 387–403, 1997.
- [19] D. Zhou, Z. Wang, and M. Schwager, “Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916–923, 2018.
- [20] J. Wang and M. Xin, “Integrated optimal formation control of multiple unmanned aerial vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1731–1744, 2012.
- [21] Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon, “Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller,” *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1827–1832, 2011.
- [22] D. Panagou and V. Kumar, “Cooperative visibility maintenance for leader-follower formations in obstacle environments,” *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 831–844, 2014.
- [23] H. Xiao, Z. Li, and C. P. Chen, “Formation control of leader-follower mobile robots’ systems using model predictive control based on neural-dynamic optimization,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5752–5762, 2016.
- [24] F. Xiao, L. Wang, J. Chen, and Y. Gao, “Finite-time formation control for multi-agent systems,” *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.
- [25] Z. Du, S. Wang, X. Wang, and J. Chen, “Formation-aware uav network self-organization with game-theoretic distributed topology control,” *IEEE Transactions on Cognitive Communications and Networking*, 2025.
- [26] L. Krick, M. E. Broucke, and B. A. Francis, “Stabilisation of infinitesimally rigid formations of multi-robot networks,” *International Journal of control*, vol. 82, no. 3, pp. 423–439, 2009.
- [27] S. Zhao and D. Zelazo, “Bearing rigidity and almost global bearing-only formation stabilization,” *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1255–1268, 2015.
- [28] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [29] P. Peng, W. Dong, G. Chen, and X. Zhu, “Obstacle avoidance of resilient uav swarm formation with active sensing system in the dense environment,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10 529–10 535.
- [30] W. Liu, J. Hu, H. Zhang, M. Y. Wang, and Z. Xiong, “A novel graph-based motion planner of multi-mobile robot systems with formation and obstacle constraints,” *IEEE Transactions on Robotics*, vol. 40, pp. 714–728, 2023.
- [31] R. Sharma, T. Weiss, and M. Kallmann, “Formation-aware planning and navigation with corridor shortest path maps,” in *Computer Graphics Forum*, vol. 43, no. 1. Wiley Online Library, 2024, p. e14995.
- [32] Z. Sui, Z. Pu, J. Yi, and S. Wu, “Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2358–2372, 2020.
- [33] Y. Xie, C. Yu, H. Zang, F. Gao, W. Tang, J. Huang, J. Chen, B. Xu, Y. Wu, and Y. Wang, “Multi-uav behavior-based formation with static and dynamic obstacles avoidance via reinforcement learning,” *arXiv preprint arXiv:2410.18495*, 2024.
- [34] Y. Yan, X. Li, X. Qiu, J. Qiu, J. Wang, Y. Wang, and Y. Shen, “Relative distributed formation and obstacle avoidance with multi-agent reinforcement learning,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1661–1667.
- [35] Y. Zhao and P. A. Vela, “Good feature matching: Toward accurate, robust vo/vslam with low latency,” *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 657–675, 2020.
- [36] M. Shamaiah, S. Banerjee, and H. Vikalo, “Greedy sensor selection: Leveraging submodularity,” in *49th IEEE conference on decision and control (CDC)*, 2010, pp. 2572–2577.
- [37] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, “Resilient flocking for mobile robot teams,” *IEEE Robotics and Automation letters*, vol. 2, no. 2, pp. 1039–1046, 2017.
- [38] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, “Formations for resilient robot teams,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 841–848, 2017.
- [39] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, “Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1261–1285, 2016.
- [40] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, “3d registration with maximal cliques,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 745–17 754.
- [41] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and

- automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [42] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pair-wise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 2916–2923.
- [43] P. J. Huber and E. M. Ronchetti, *Robust statistics*. John Wiley & Sons, 2011.
- [44] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.
- [45] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.
- [46] H. G. de Marina, "Maneuvering and robustness issues in undirected displacement-consensus-based formation control," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3370–3377, 2020.
- [47] L. Asimow and B. Roth, "The rigidity of graphs, ii," *Journal of Mathematical Analysis and Applications*, vol. 68, no. 1, pp. 171–190, 1979.
- [48] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *Journal of Combinatorial Theory, Series B*, vol. 94, no. 1, pp. 1–29, 2005.
- [49] M. Tantardini, F. Ieva, L. Tajoli, and C. Piccardi, "Comparing methods for comparing networks," *Scientific reports*, vol. 9, no. 1, p. 17557, 2019.
- [50] K.-t. Leung, *Linear algebra and geometry*. Hong Kong University Press, 1974, vol. 1.
- [51] D. Garamvölgyi and T. Jordán, "Minimally globally rigid graphs," *European Journal of Combinatorics*, vol. 108, p. 103626, 2023.
- [52] T. Jordán and S.-i. Tanigawa, "Globally rigid powers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 155, pp. 111–140, 2022.
- [53] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1, 2022, pp. 172–183.
- [54] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.
- [55] S. T. Jawaaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.
- [56] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2015.
- [57] V. Batagelj and M. Zaversnik, "An $\mathcal{O}(m)$ algorithm for cores decomposition of networks," *arXiv preprint cs/0310049*, 2003.
- [58] J. L. Walteros and A. Buchanan, "Why is maximum clique often easy in practice?" *Operations Research*, vol. 68, no. 6, pp. 1866–1895, 2020.
- [59] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.
- [60] L. S. Jennings and K. L. Teo, "A computational algorithm for functional inequality constrained optimization problems," *Automatica*, vol. 26, no. 2, pp. 371–375, 1990.
- [61] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on $\mathbb{SE}(3)$," in *49th IEEE conference on decision and control (CDC)*, 2010, pp. 5420–5425.
- [62] 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," *IEEE Transactions on Robotics*, 2025.
- [63] K. Kondo, J. Tordesillas, R. Figueroa, J. Rached, J. Merkel, P. C. Lusk, and J. P. How, "Robust mader: Decentralized and asynchronous multiagent trajectory planner robust to communication delay," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1687–1693.
- [64] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. P. Chen, "Review of advanced guidance and control algorithms for space/aerospace vehicles," *Progress in Aerospace Sciences*, vol. 122, p. 100696, 2021.