Subteaming and Adaptive Formation Control for Coordinated Multi-Robot Navigation

Zihao Deng[1*], Peng Gao[2*], Williard Joshua Jose[1], Maggie Wigness[3], John Rogers[3], Brian Reily[3], Christopher Reardon[4], and Hao Zhang[1]

[1] University of Massachusetts Amherst [2] North Carolina State University
[3] U.S. Army DEVCOM Army Research Laboratory [4] University of Denver
[*] Authors contributed equally to this paper

Abstract: Coordinated multi-robot navigation is essential for robots to operate as a team in diverse environments. During navigation, robot teams usually need to maintain specific formations, such as circular formations to protect human teammates at the center. However, in complex scenarios such as narrow corridors, rigidly preserving predefined formations can become infeasible. Therefore, robot teams must be capable of dynamically splitting into smaller subteams and adaptively controlling the subteams to navigate through such scenarios while preserving formations. To enable this capability, we introduce a novel method for *SubTeaming and Adaptive Formation* (STAF), which is built upon a unified hierarchical learning framework: (1) high-level deep graph cut for team splitting, (2) intermediate-level graph learning for facilitating coordinated navigation among subteams, and (3) low-level policy learning for controlling individual mobile robots to reach their goal positions while avoiding collisions. To evaluate STAF, we conducted extensive experiments in both indoor and outdoor environments using robotics simulations and physical robot teams. Experimental results show that STAF enables the novel capability for subteaming and adaptive formation control, and achieves promising performance in coordinated multi-robot navigation through challenging scenarios. More details are available on the project website: https://hcrlab.gitlab.io/project/STAF.

Keywords: Coordinated multi-robot navigation, subteam, hierarchical learning.

1 Introduction

Multi-robot systems have attracted growing attention due to their advantages, such as redundancy [1], parallelism [2], and scalability [3]. Coordinated multi-robot navigation is a fundamental capability that allows teams of robots to traverse environments in a synchronized manner and reach goal positions collectively [4]. This capability is crucial in real-world applications, such as search and rescue [5, 6, 7], space exploration [8, 9], and transportation [10, 11].

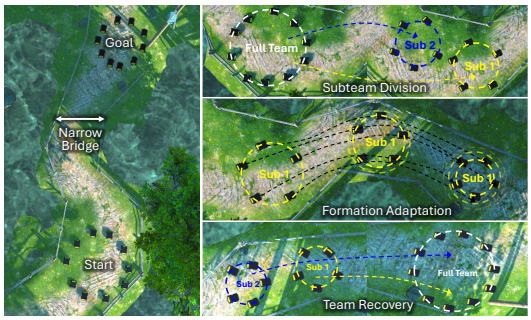


Figure 1: When a robot team in circular formation encounters a bridge that is too narrow for the entire team to cross at once. The robots must divide into subteams, adapt their formations to navigate the bridge, and recovery the full team after crossing.

During coordinated navigation, robots are often required to maintain mission-specific formation, such as circular formations for protection or line formations for coverage. However, rigid adherence to predefined formations can hinder effective navigation in certain scenarios. For instance, Figure 1 depicts a team of ten robots in a circular

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

formation encountering a corridor too narrow for the entire team to pass through. Thus, the team must be capable of dynamically dividing into smaller units that operate both independently and cohesively (i.e., *subteaming*) and controlling the subteams to pass through the narrow corridor while adaptively maintaining a specific formation (i.e., *adaptive formation control*).

The importance of coordinated multi-robot navigation has driven the development of various techniques. Traditional approaches, including classical planning methods [12], game-theoretic methods [13, 14], and optimization-based methods [15], often face high computational costs. Recently, learning-based methods like deep neural networks [16, 17] and multi-agent reinforcement learning [18, 19] have been used for modeling, coordination, and navigation. However, these methods have not addressed adaptive formation control, which is critical for narrow corridor traversal. Subteaming methods, such as graph cuts for team division [20, 21] and mixed-integer programming for task allocation [1, 22, 23], focus on team division alone and lack control over subteams or individual robots, which limits their effectiveness for coordinated navigation.

To address the challenges above and enable effective coordinated multi-robot navigation in complex scenarios where the entire robot team cannot pass through, we introduce a novel approach called *SubTeaming and Adaptive Formation* (**STAF**), which offers new capabilities for subteam division, formation adaptation, and team recovery. Specifically, we design a graph representation to encode a team of robots, where each node represents a robot along with its associated attributes, such as its position, velocity, goal, and distance to obstacles, and each edge represents the spatial relationships between pairs of robots. Our STAF approach integrates three levels of robot learning into a hierarchical framework. At the high level, given the graph representation of a robot team, STAF performs deep graph cuts to divide the entire robot team into subteams. The intermediate level of STAF focuses on learning the coordination of these robot subteams for navigation, which develops a graph neural network with learnable message sharing to coordinate robots within a subteam, while generating graph embeddings to encode the subteam context. Finally, at the low level, given these embeddings, STAF employs reinforcement learning to learn a navigation policy that controls each individual robot to adaptively maintain subteam formation, reach the goal position, and avoid collisions.

Our primary contribution is the introduction of the novel STAF method to enable a new multi-robot navigation capability of subteaming and formation adaptation. The specific novelties include:

- This work introduces one of the first problem formulations and learning-based solutions for subteaming and formation adaptation in multi-robot coordinated navigation. It enables new multi-robot capabilities, including subteam division, formation adaptation, and team recovery, allowing a team of robots to navigate complex environments in a coordinated manner, particularly narrow corridors where maintaining original formation is infeasible.
- We introduce a novel hierarchical robot learning method that simultaneously integrates high-level deep graph cut for subteaming, intermediate-level graph learning for subteam coordination and adaptive formation control, and low-level individual robot control for collision-free navigation in complex environments.

2 Related Work

Hierarchical Learning for Robotics Hierarchical learning has shown promise in complex multi-robot tasks by providing a structured problem formulation that better aligns with multi-objective goals. It also enhances modularity in model design, which improves interpretability and enables clearer evaluation of each level. Applications include task allocation [24], maintaining communication [25], path planning [26, 27], and consensus reaching [28]. Typically, the lower level handles individual control tasks such as obstacle avoidance [29, 30]. The upper level focuses on team planning and coordination [31, 32, 33, 18]. However, applying hierarchical learning to formation adaptation and subteaming remains challenging due to the need for scalable team representations, dynamic adaptation, and efficient integration of formation control with flexible team reconfiguration.

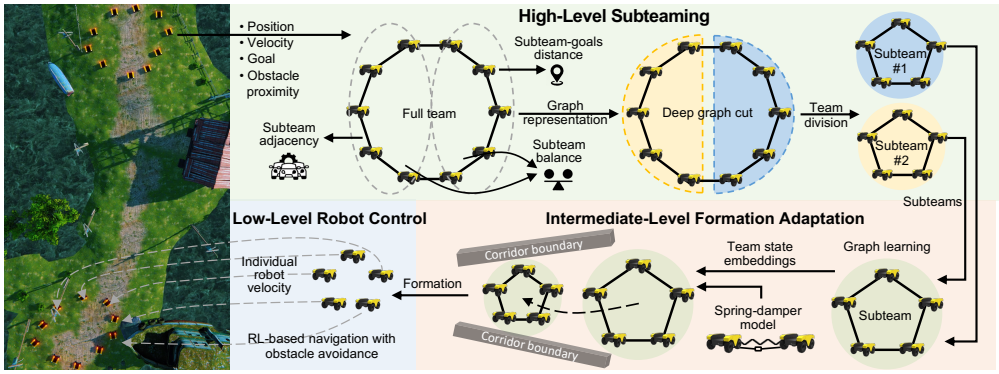


Figure 2: Overview of STAF, which integrates three levels of robot learning within a unified hierarchical learning framework to enable coordinated multi-robot navigation.

Coordinated Multi-Robot Navigation Learning-free methods rely on predefined formation strategies, such as leader-follower [15, 4, 34, 35] and virtual region methods [36, 37, 38, 39]. However, these rigid formations lack adaptability to environmental changes. Learning-based methods, such as reinforcement learning (RL) [40, 41, 18, 19, 42, 41], address this limitation by optimizing actions through environmental feedback. Graph neural networks (GNNs) enhance team coordination and communication [43, 17], enabling decentralized decision-making [16, 44]. These approaches have been applied in areas such as connected autonomous driving [45, 8], area coverage [46], and search-and-rescue missions [5]. However, none of these methods effectively address subteaming and formation adaptation in coordinated navigation, particularly in complex narrow corridors.

Subteaming in Multi-Robot Navigation and Task Allocation Subteaming increases the complexity of coordinated multi-robot navigation as it involves splitting, merging, and reformation based on tasks or environments. Graph-based methods [20, 21, 47, 1] use graph partitioning and cutting to determine team division and merging, but often rely on explicit connectivity constraints. Leader-follower methods [48, 49, 15] apply predefined hierarchy-based strategies but lack flexibility in dynamic environments. Optimization-based approaches [50, 22, 23, 51] compute optimal assignments via mixed-integer programming. Heuristic-based methods [52, 53] use problem-specific heuristics to determine team formation and coordination strategies. However, these methods focus on team division alone and lack control over subteams or individual robots. See Appendix A for details [1].

3 Approach

Problem Definition We discuss our STAF method that enables new multi-robot capabilities of subteaming and formation adaptation for coordinated multi-robot navigation. An overview of STAF is illustrated in Figure 2. We represent a team of n robots using an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathbf{E}\}$. In the node set $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, each node $\mathbf{v}_i = \{\mathbf{p}_i, \mathbf{g}_i, \mathbf{q}_i\}$ consists of the attributes of the i-th robot, where $\mathbf{p}_i = [p_i^x, p_i^y]$ denotes its position, $\mathbf{g}_i = [g_i^x, g_i^y]$ denotes its goal position, and $\mathbf{q}_i = [q_i^x, q_i^y]$ denotes its velocities along x and y directions. The edge matrix $\mathbf{E} = \{a_{i,j}\}^{n \times n}$ represents the spatial adjacency of the robots, where $a_{i,j} = 1$, if the i-th robot and the j-th robot are within a radius; otherwise $a_{i,j} = 0$. We further define the state of the i-th robot $\mathbf{s}_i = [\mathbf{p}_i, \mathbf{g}_i, \mathbf{q}_i, c_i]$ as the concatenation of the robot's attributes and the distance c_i between the robot and its closest obstacle. We define the action of the i-th robot as $\mathbf{a}_i = [v_i^x, v_i^y]$, where v_i^x and v_i^y denote the robot's velocities in the x and y directions, respectively.

Our objective is to address the problems of subteaming and formation adaptation in the context of coordinated multi-robot navigation:

[1]The appendix is available at our project website.

- **Formation Adaptation:** The capability of a robot team or subteam to maintain a desired formation while dynamically adjusting their relative positions to safely and efficiently navigate through the unstructured environment toward their goal positions, particularly in challenging scenarios such as narrow corridors.
- **Subteaming**: The capability of a robot team with a specific formation to autonomously divide into subteams with the same formation type when navigating environments too narrow for the entire robot team. After successfully passing through, the subteams must merge back into the full team, restoring the original formation.

High-Level Deep Graph Cut for Subteaming Given the graph $\mathcal{G}$ as the representation of the robot team, we introduce a new deep graph cut approach at the high level of STAF to enable subteaming. We compute the embedding of the robot graph as $\mathcal{H} = \{\mathbf{h}_i\} = \omega(\mathcal{G})$, where $\mathbf{h}_i$ is the embedding of the i-th robot and ω is a graph attention network [54]. We project each node into a representation space by calculating $\mathbf{m}_i = \mathbf{W}^v \mathbf{p}_i$, where $\mathbf{m}_i$ denotes the projected feature vector of the i-th node, and $\mathbf{W}^v$ denotes the weight matrix. Then, we compute the attention $\alpha_{i,j}$ from the j-th node to the i-th node as $\alpha_{i,j} = \frac{\exp(\text{ReLu}([\mathbf{W}^a \mathbf{m}_i || \mathbf{W}^a \mathbf{m}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{ReLu}([\mathbf{W}^a \mathbf{m}_i || \mathbf{W}^a \mathbf{m}_k]))}$, where ReLu denotes the rectified linear unit activation function, $\mathcal{N}(i)$ represents the set of adjacent nodes of the i-th node, $||$ denotes the concatenation operation, and $\mathbf{W}^a$ represents the weight matrix. The attention $\alpha_{i,j}$ is obtained by computing the similarity of the i-th node with its j-th adjacent nodes, followed by the SoftMax normalization. Then, the final embedding $\mathbf{h}_i$ for the i-th node is computed through aggregating the embeddings of all its adjacent nodes as $\mathbf{h}_i = \mathbf{W}^h \mathbf{m}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \mathbf{W}^h \mathbf{m}_j$, where $\mathbf{W}^h$ is the weight matrix. We further utilize a multi-head mechanism [54] after the attention layers to enable the network to capture a richer embedding representation.

Given $\mathcal{H} = \{\mathbf{h}_i\}$, we formulate subteaming as a graph cut problem, which partitions the entire graph (representing the full team) into m subgraphs (representing subteams). In order to compute team division, we develop a classifier network $\tau(\mathcal{H})$ consisting of two fully connected linear layers followed by a SoftMax function, which outputs the team division results as $\mathbf{Y} = \tau(\mathcal{H}) = \{y_{i,j}\}^{n \times m}$, where $y_{i,j}$ is the probability of the i-th robot belonging to the j-th subteam, and $m < n$.

To ensure that robots within the same subteam group together, i.e., each robot is adjacent to its teammates within the same subteam, we define a loss function that maximizes the adjacency of robots within each subteam as $\mathbf{Y}(1-\mathbf{Y})^\top \mathbf{E}$, where $\mathbf{Y}(1-\mathbf{Y})^\top$ calculates the probability that a pair of robots belong to different subteams, and $\mathbf{E}$ encodes the adjacency of the robots. In addition, we aim to maintain balance in the sizes of robot subteams, encouraging each subteam to have the same or a similar number of robots. It can be mathematically modeled by a loss function $\sum_{j=1}^{m} \left(\sum_{i=1}^{n} y_{i,j} - \frac{n}{m} \right)$. The term $\frac{n}{m}$ calculates the optimal size of balanced subteams (e.g., when $n = 10$ and $m = 2$, each subteam would consist of 5 robots). Furthermore, we model the mission objective of reaching the goal position by minimizing the overall distance between the subteams and their respective goal positions. It can be mathematically defined as $\sum_{j=1}^{m} \left\| \frac{\sum_{i=1}^{n} y_{i,j} \mathbf{p}_i}{\sum_{i=1}^{n} y_{i,j}} - \frac{\sum_{i=1}^{n} y_{i,j} \mathbf{g}_i}{\sum_{i=1}^{n} y_{i,j}} \right\|_2$, where $\frac{\sum_{i=1}^{n} y_{i,j} \mathbf{p}_i}{\sum_{i=1}^{n} y_{i,j}}$ denotes the center position of the j-th subteam and $\frac{\sum_{i=1}^{n} y_{i,j} \mathbf{g}_i}{\sum_{i=1}^{n} y_{i,j}}$ denotes the center position of the goal for the subteam.

The high-level component of STAF performs an unsupervised graph cut to enable team division for subteaming by minimizing the following objective function:

$$\mathcal{L}_{st} = \overbrace{\mathbf{Y}(1-\mathbf{Y})^\top \mathbf{E}}^{\text{Subteam adjacency}} + \overbrace{\sum_{j=1}^{m} \left(\sum_{i=1}^{n} y_{i,j} - \frac{n}{m} \right)}^{\text{Subteam balance}} + \overbrace{\sum_{j=1}^{m} \left\| \frac{\sum_{i=1}^{n} y_{i,j} \mathbf{p}_i}{\sum_{i=1}^{n} y_{i,j}} - \frac{\sum_{i=1}^{n} y_{i,j} \mathbf{g}_i}{\sum_{i=1}^{n} y_{i,j}} \right\|_2}^{\text{Subteam-goals distance}} \tag{1}$$

which jointly accounts for subteam adjacency, subteam balance, and subteam-goal distances.

Intermediate-Level Graph Learning for Multi-robot Formation Adaptation To enable adaptive multi-robot formation control, we develop a graph learning approach at the intermediate level of STAF, which coordinates multiple robots to maintain a specific formation while adapting it based

on the surrounding environment. Given $\mathcal{G}$ that represents a team (or subteam) of robots along with the state $\mathbf{s}_i$ for each robot i, we develop a graph network ϕ to compute the embedding $\mathbf{f}_i = \phi(\mathbf{s}_i, \mathcal{G})$ of the team state with respect to the i-th robot, which encodes the spatial relationships between the i-th robot with others in the team. The network ϕ uses a linear layer to project the robot state $\mathbf{s}_i$ to the individual embedding $\mathbf{z}_i$ of the i-th robot by $\mathbf{z}_i = \mathbf{W}^z \mathbf{s}_i$, where $\mathbf{W}^z$ is the weight matrix of the linear layer. Then, for the i-th robot, ϕ aggregates individual state embeddings of all other teammates through message passing to compute the team state embedding $\mathbf{f}_i$ with respect to the i-th robot as $\mathbf{f}_i = \mathbf{W}^f \mathbf{z}_i + \sum_{j \in \mathcal{N}(i)} \mathbf{W}^f (\mathbf{z}_j - \mathbf{z}_i)$, where $\mathbf{W}^f$ is the weight matrix. The team state embedding $\mathbf{f}_i$ with respect to the i-th robot encodes not only its own states (captured in the first term), but also the relative spatial relationships with other teammates (captured in the second term), which facilitates the coordination of actions to maintain specific formations during multi-robot navigation.

Robot teams and subteams may encounter scenarios, such as narrow corridors, where rigidly maintaining their formations prevents successful navigation. To enable formation adaptation, we implement a spring-damper model [55, 56] that dynamically adjusts the shape of the formation within the same type. This spring-damper model includes two components: (1) The spring component ensures that robot pairs maintain a balance between staying close enough to navigate narrow corridors and keeping a sufficient distance to prevent collisions, with the flexibility to adjust formation and enable adaptation. This spring component is modeled as $|d_{i,j} - p_{i,j}|$, where $d_{i,j}$ denotes the expected distance in the original formation and $p_{i,j}$ represents the actual distance between the i-th and j-th robots, computed as $\|\mathbf{p}_i - \mathbf{p}_j\|_2$. (2) The damper component prevents oscillation and overshooting of each robot during navigation by smoothing the relative velocities between pairs of robots, which is defined as $q_{i,j} = \|\mathbf{q}_i - \mathbf{q}_j\|_2$. Combining these components, the spring-damper model for formation adaptation is mathematically defined as $R^{adp} = \sum_{\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}} -\lambda |d_{i,j} - p_{i,j}| - (1-\lambda) q_{i,j}$, where λ is a hyperparameter that balances the importance of the spring and damper components. R^{adp} is incorporated into the reward function, which is used to derive a loss function for training STAF.

Low-Level Individual Robot Control for Navigation At the low-level of STAF, we introduce a navigation control network that outputs velocity commands as actions for each robot to reach its goal. Given the state $\mathbf{s}_i$ for the i-th robot, we compute its state embedding $\mathbf{f}_i$. We design the network ψ, which consists of two linear layers followed by the ReLU activation function, maps this embedding to an action as $\mathbf{a}_i = \psi(\mathbf{f}_i)$. The network ψ is a part of the control policy $\pi_\theta(\mathbf{a}_i|\mathbf{s}_i)$, parameterized by θ, which is trained using the framework of reinforcement learning. To enable each robot to move toward its target position and reach the navigational goal, we design a reward function based upon the distance between the current positions of the robot and its goal position. To enable obstacle avoidance for safe navigation, we implement a reward function that imposes a penalty when a robot comes too close to nearby obstacles or other robots in the team. When robots are divided into subteams, and once all subteams pass through the narrow corridor into an open area that is large enough for the full team, the goal position of each individual robot is updated to align with the full team's goal, thereby recovering the subteams back into the full team with the original formation.

See Appendix B for details on **STAF Training and Execution** with their time complexity analysis.

4 Experiments

Experimental Setups We comprehensively evaluate our STAF approach across three setups: (1) a standard Gazebo simulation in ROS1, (2) a high-fidelity Unity-based 3D multi-robot simulator in ROS1, and (3) physical robot teams running ROS2. Each setup involves different numbers and types of robots arranged in formations such as circle, wedge, and line. In all scenarios, the environment includes narrow corridors, which require the full robot team to divide into subteams that adapt their formation to pass through. Afterward, the subteams regroup into the original full-team formation. In simulation, robot poses and obstacles are obtained from Gazebo and Unity. In real-world experiments, robots use a SLAM approach [57] for state estimation and mapping. See Appendix C for details on approach implementation and training. All video demonstrations are available on our project website.

Table 1: Quantitative comparison of STAF and Previous Methods from Gazebo simulations in ROS1.

Method	Circle Formation					Wedge Formation					Line Formation				
	SR (%)	TT (sec)	$\sigma < 0.5$	$\sigma < 0.1$	$\sigma < 0.01$	SR (%)	TT (sec)	$\sigma < 0.5$	$\sigma < 0.1$	$\sigma < 0.01$	SR (%)	TT (sec)	$\sigma < 0.5$	$\sigma < 0.1$	$\sigma < 0.01$
DGNN [18]	**100.00**	68.70	60.41	58.91	58.91	**100.00**	82.70	47.85	42.33	41.92	**100.00**	72.61	27.90	20.16	20.16
L&F [15]	40.00	**27.40**	67.28	64.54	62.69	70.00	**26.50**	69.70	62.11	59.47	60.00	**30.10**	63.76	55.89	55.89
STAF-full	**100.00**	102.10	**87.79**	**80.12**	**80.12**	**100.00**	69.30	**80.52**	**80.51**	**80.50**	**100.00**	111.50	**91.45**	**80.06**	**78.93**

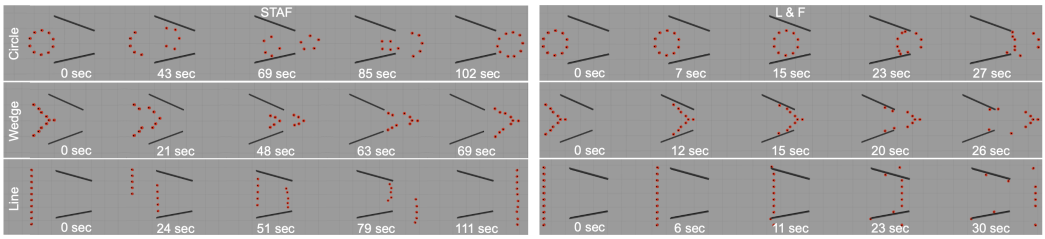
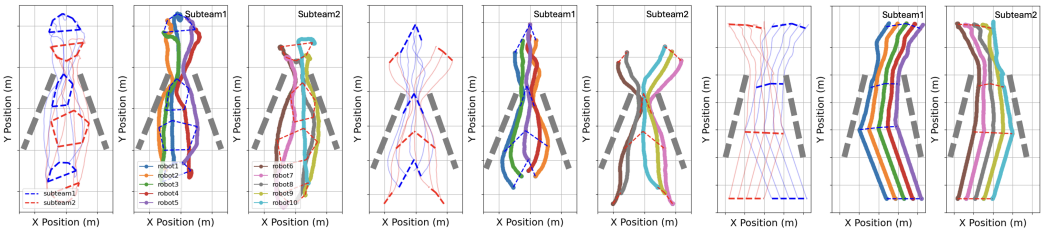


Figure 3: Qualitative results from Gazebo simulations on subteaming and formation adaptation.



(a) Circle formation (b) Wedge formation (c) Line formation

Figure 4: Movement trajectories of ten robots navigating a narrow corridor with different formations. In Figure 4(a) to 4(c), the first subfigure displays two subteams (red and blue) during team division, navigation with formation adaptation, and regrouping. The second and third subfigures show subteam trajectories, with each robot's path in a distinct color and gray dashed lines indicating obstacles.

We implement the complete STAF approach referred to as **STAF-full**. The full team divides into subteams to navigate through narrow environments, and after passing through, the subteams regroup into the full team to its original formation. To analyze the performance of the subteams, we refer to the subteams as **STAF-sub#**, e.g., STAF-sub1 and STAF-sub2. For comparison with STAF, we further implement two previous methods for multi-robot coordinated navigation, including: (1) A Leader and Follower method (**L&F**) [15] that one of the robots is designated as the "leader robot" that leads the movements of the other "follower robots" in the team while maintaining the formation. (2) Decentralized GNN (**DGNN**) [18] that built upon a hierarchical learning framework to generate velocity controls for each individual robot for navigation, without considering team-level formations.

To quantitatively evaluate and compare with other methods, we employ three metrics, including: (1) Successful Rate (**SR**) is defined as the proportion of the robots within the full team that successfully reach goal positions without collisions. (2) Travel Time (**TT**) is defined as the total time used by the full team to reach the goal position. (3) Contextual Formation Integrity (**CFI**) is defined as the real-time adherence of the robots to their designated formation, given a shape threshold σ that defines the strictness of the formation. The CFI $\in [0, 1]$ evaluates how effectively a robot team utilizes the corridor gap and maintains formation, with smaller σ indicating stricter formation requirements. See Appendix D for details on CFI and its calculation of different formations.

Results in Multi-Robot Simulations The qualitative results in the Gazebo simulation are shown in Figure 3. L&F gets stuck in the narrow corridor due to the lack of subteaming and formation adaptation. In contrast, our method autonomously divides the team, enabling each subteam to adapt formations and reach the goal; the first subteam starts moving, followed by the second, and they eventually merge into the full formation. Notably, for wedge formations, team division prioritizes goal-distance objectives instead of maximizing connectivity, resulting in more compact subteams.

We visualize the trajectories of a team of 10 robots navigating in different formations, as shown in Figure 4. The visualization reveals subteaming behaviors (indicated by subteams in red and blue colors), including team division and regrouping. Additionally, formation adaptation of each subteam

occurs when navigating through narrow corridors (indicated by the individual robot trajectories). These results show the effectiveness of STAF in enabling both subteaming and formation adaptation.

The quantitative results are shown in Table 1. DGNN performs the worst, particularly in the CFI metrics, as it lacks formation control. L&F uses formation control and performs better but has only a 40% success rate, as it considers rigid formation control. Our method outperforms both by addressing these limitations, which achieves a 100% success rate. Although STAF yields slightly longer travel times, this is expected due to its more complex navigation strategy. Subteam performance in Table 2 shows a 100% success rate across all formations. STAF maintains formation integrity above 87%, 80%, and 91% under the threshold $\sigma < 0.5$ for circle, wedge, and line formations, respectively. These results highlight the effectiveness of STAF in enabling coordinated navigation through subteaming and adaptive formation control.

Table 2: Quantitative results of two subteams from Gazebo simulations in ROS1.

Subteam	Formation	Metrics				
		SR (%)	TT (sec)	$\sigma < 0.5$	$\sigma < 0.1$	$\sigma < 0.01$
STAF-sub1	Circle	100.00	84.80	81.56	71.69	70.59
	Wedge	100.00	58.80	77.22	71.79	69.86
	Line	100.00	78.51	91.13	79.53	77.43
STAF-sub2	Circle	100.00	59.50	87.72	82.67	80.11
	Wedge	100.00	46.80	80.99	80.15	75.79
	Line	100.00	90.06	91.78	81.74	80.43

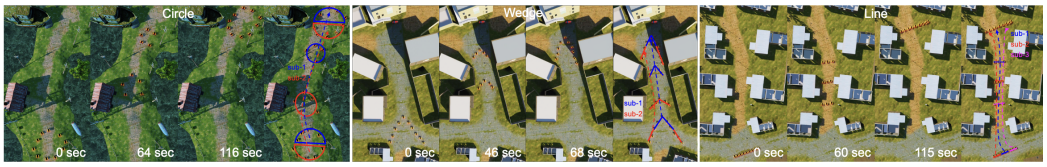


Figure 5: Qualitative results from Unity3D simulations in ROS1 using varying numbers of differential-drive Warthog robots in three formations while navigating a long, unstructured field environment.

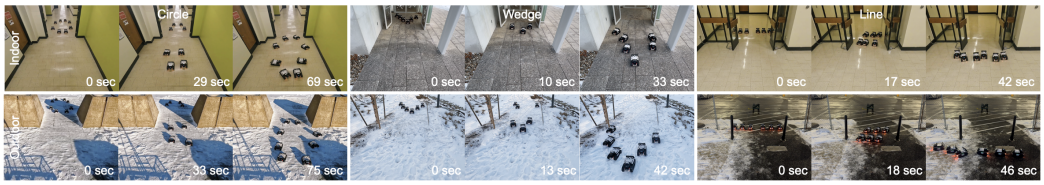


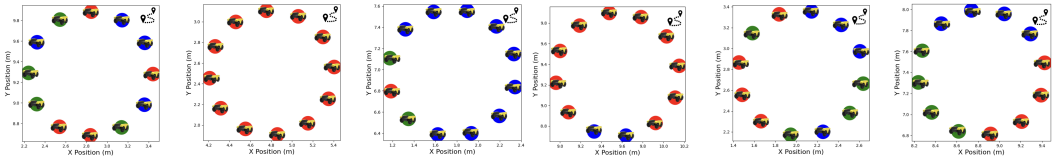
Figure 6: Qualitative results from real-world experiments in both indoor narrow spaces and outdoor uneven terrain, using varying numbers of Limo robots running ROS2 and communicating via Wi-Fi.

Beyond the Gazebo simulation, we further use a high-fidelity Unity3D simulator in ROS1, which simulates outdoor field environments with narrow pathways and bridges. Instead of using holonomic robots as shown in the Gazebo simulation, we use differential-drive Warthog robots and convert the linear velocity in the action $\mathbf{a}_i$ into wheel velocities to follow the same trajectory. This setting introduces new challenges, which require the robot team to navigate long curved paths that demand continuous formation adjustments. As illustrated in Figures 5, our STAF approach successfully addresses these challenges by dividing a full team into subteams, adapting actions of differential-drive subteams to navigate, and regroup after subteam traversal. For line formation with 9 robots, STAF can divide into three subteams to navigate a corridor too narrow for groups larger than 3.

Case Study on Physical Robot Teams We validate STAF on real-world case studies using differential-drive Limo robots with caterpillar tracks, each equipped with an onboard Intel NCU i7 and running ROS2 with Wi-Fi-based team communication. The real-world experiments are conducted both indoors and outdoors, as shown in Figure 6. Our method enables teams of 6 to 8 robots to divide into subteams and adapt formations to smoothly navigate narrow indoor spaces. In outdoor experiments on unstructured terrain, the results demonstrate the strong adaptability of our approach to unknown environments. Subteaming and formation adaptation are effectively performed even on snowy and uneven terrain, where wheel slippage introduces significant action uncertainty. Additional Unity3D and real-world qualitative results with more timesteps are provided in the Appendix E.

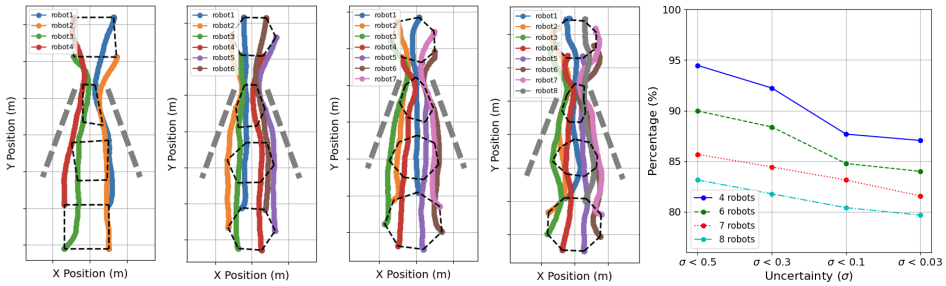
5 Discussion

Ablation Study on Subteam Division We conduct an ablation study to evaluate the role of each component in the objective function defined in Eq. (1) for team division. Figure 7(a) shows that



(a) ST-B (b) ST-A (c) ST-G (d) w/o ST-B (e) w/o ST-A (f) w/o ST-G

Figure 7: Ablation study that analyzes the impact of subteam division components: subteam balance (ST-B), subteam adjacency (ST-A), and subteam-goals distance (ST-G).



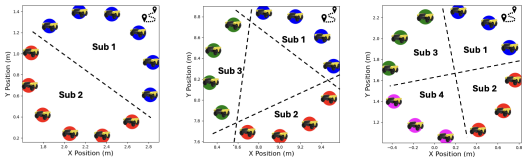
(a) 4 robots (b) 6 robots (c) 7 robots (d) 8 robots (e) CFI across team sizes

Figure 8: Quantitative results indicate STAF's generalizability to different team sizes. Figures (a)-(d) show the trajectories of 4 to 8 robots in circle formations to navigate a narrow corridor. Figure (e) presents the variation in CFI values across different team sizes and σ values.

optimizing only the balance term evenly splits 12 robots into 3 subteams. Figure 7(b) shows that only maximizing adjacency leads to all robots being assigned to the same subteam. Figure 7(c) shows that only minimizing the goal-distance aligns subteams toward their goals (in the upper right). In addition, we remove each component individually to assess its impact. Figure 7(d) shows unbalanced team division without the balance term. Figure 7(e) results in uncompact subteams without the adjacency term. Figure 7(f) shows subteams misaligned with goals, which leads to inefficient navigation. These results further indicate the effectiveness and importance of enforcing subteam balance, maximizing adjacency, and minimizing subteam-goals distance for robot team division.

Generalizability to Different Team Sizes We evaluate the generalizability of STAF to different team sizes by varying the number of robots. Figures 8(a)-8(d) present the qualitative results on formation adaptation for teams of 4, 6, 7, and 8 robots in circle formation, which validate STAF's generalizability across team sizes. Figure 8(e) presents the quantitative results using the CFI metric, which shows 87% formation integrity for 4 robots under $\sigma < 0.03$, and at least 80% for 8 robots.

Generalizability to Different Numbers of Subteams We evaluate STAF's generalizability in dividing the team into varying numbers of subteams. As shown in Figure 9, STAF effectively handles divisions into 2, 3, and 4 subteams. Figure 5 contains a scenario where a nine-robot line formation splits into three subteams to navigate a corridor too narrow for groups larger than four.



(a) 2 subteams (b) 3 subteams (c) 4 subteams

Figure 9: Qualitative results indicate STAF's generalizability to different numbers of subteams.

See Appendix F for STAF's **Robustness to Noise** and **Applicability to Different Robot Platforms**.

6 Conclusion

In this paper, we propose STAF for coordinated multi-robot navigation in complex scenarios. STAF is built upon a unified hierarchical learning framework, including a high-level deep graph cut for dynamic team division, an intermediate-level graph learning for team coordination with adaptive formation control, and a low-level RL policy for individual robot control. Results from comprehensive experiments show that STAF enables new multi-robot capabilities for subteaming and formation adaptation, and significantly outperforms existing methods on coordinated multi-robot navigation.

7 Limitations

Our approach presents several limitations that suggest directions for future research. First, although STAF's intermediate and low levels are executed in a decentralized fashion, STAF's high level for team division is executed in a centralized fashion. One direction for future research is to decentralize the high-level team division, such as by replacing the current global graph cut optimization with a distributed consensus algorithm (e.g., gossip [58] or max-consensus [59]). These decentralized methods would enable each robot to determine its subteam based upon the information shared by its teammates through broadcasting, and iteratively reach a consensus and converge to a stable subteam assignment through negotiation. Second, the alternating training algorithm we use, which iteratively trains the high-level and joint intermediate-low levels, is considered a limitation, as it may lead to suboptimal integration of these levels and difficulties with training error propagation. In the future, we plan to integrate the high-level graph cut together with the joint intermediate-low level training into an end-to-end training algorithm, where the training error from the low level will be propagated not only to the intermediate level but also to the high level, which enables updates to the network parameters across all three levels. To achieve this, we will adopt a centralized training with decentralized execution strategy, where all levels of the hierarchy can leverage global information during training, while ensuring decentralized execution during deployment. The third limitation is that the number of subteams, as a hyperparameter, is decided manually. A future direction is to dynamically and adaptively determine this hyperparameter by selecting the minimum number of subteams such that the smallest formation of each subteam can successfully navigate through the narrowest corridor in the environment. The width of a corridor can be identified either by analyzing the environment map (using a prior map or built by a SLAM method) or through real-time robotic sensing.

Acknowledgments

This work was partially supported by the NSF CAREER Award IIS-2308492, DEVCOM ARL TBAM CRA W911NF2520024, and the DARPA Young Faculty Award (YFA) D21AP10114-00.

References

[1] P. Gao, S. Siva, A. Micciche, and H. Zhang. Collaborative scheduling with adaptation to failure for heterogeneous robot teams. In *IEEE International Conference on Robotics and Automation*, 2023.

[2] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, et al. Argos: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6:271–295, 2012.

[3] T. Balch and M. Hybinette. Social potentials for scalable multi-robot formations. In *IEEE International Conference on Robotics and Automation*, 2000.

[4] P. Singh, R. Tiwari, and M. Bhattacharya. Navigation in multi robot system using cooperative learning: A survey. In *International Conference on Computational Techniques in Information and Communication Technologies*, 2016.

[5] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8:191617–191643, 2020.

[6] Q. Yang and R. Parasuraman. Needs-driven heterogeneous multi-robot cooperation in rescue missions. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2020.

[7] J. L. Baxter, E. Burke, J. M. Garibaldi, and M. Norman. Multi-robot search and rescue: A potential field based approach. *Autonomous Robots and Agents*, pages 9–16, 2007.

[8] R. Han, S. Chen, and Q. Hao. Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2020.

[9] V. Indelman. Cooperative multi-robot belief space planning for autonomous navigation in unknown environments. *Autonomous Robots*, 42:353–373, 2018.

[10] A. Amanatiadis, C. Henschel, B. Birkicht, B. Andel, K. Charalampous, I. Kostavelis, R. May, and A. Gasteratos. Avert: An autonomous multi-robot system for vehicle extraction and transportation. In *IEEE International Conference on Robotics and Automation*, 2015.

[11] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer. Cooperative multi-robot object transportation system based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 6(4):6466–6472, 2021.

[12] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation*, 2000.

[13] B. Tang, K. Xiang, M. Pang, and Z. Zhanxia. Multi-robot path planning using an improved self-adaptive particle swarm optimization. *International Journal of Advanced Robotic Systems*, 17(5):1729881420936154, 2020.

[14] D. Cappello, S. Garcin, Z. Mao, M. Sassano, A. Paranjape, and T. Mylvaganam. A hybrid controller for multi-agent collision avoidance via a differential game formulation. *IEEE Transactions on Control Systems Technology*, 29(4):1750–1757, 2021.

[15] B. Reily, C. Reardon, and H. Zhang. Leading multi-agent teams to multiple goals while maintaining communication. In *Robotics Science and Systems*, 2020.

[16] M. Goarin and G. Loianno. Graph neural network for decentralized multi-robot goal assignment. *IEEE Robotics and Automation Letters*, 2024.

[17] S. Zhang, K. Garg, and C. Fan. Neural graph control barrier functions guided distributed collision-avoidance multi-agent control. In *Conference on Robot Learning*, 2023.

[18] J. Blumenkamp, S. Morad, J. Gielis, Q. Li, and A. Prorok. A framework for real-world multi-robot systems running decentralized GNN-based policies. In *International Conference on Robotics and Automation*, 2022.

[19] Y. Hu, J. Fu, and G. Wen. Graph soft actor–critic reinforcement learning for large-scale distributed multirobot coordination. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[20] H. Zhu, J. Juhl, L. Ferranti, and J. Alonso-Mora. Distributed multi-robot formation splitting and merging in dynamic environments. In *International Conference on Robotics and Automation*, 2019.

[21] B. Reily, C. Reardon, and H. Zhang. Representing multi-robot structure through multimodal graph embedding for the selection of robot teams. In *IEEE International Conference on Robotics and Automation*, 2020.

[22] W. J. Jose and H. Zhang. Learning for dynamic subteaming and voluntary waiting in heterogeneous multi-robot collaborative scheduling. In *IEEE International Conference on Robotics and Automation*, 2024.

[23] G. A. Cardona, K. Leahy, and C.-I. Vasile. Temporal logic swarm control with splitting and merging. In *IEEE International Conference on Robotics and Automation*, 2023.

[24] R. Wang, Z. Hua, G. Liu, J. Zhang, J. Yan, F. Qi, S. Yang, J. Zhou, and X. Yang. A bi-level framework for learning to solve combinatorial optimization on graphs. *Advances in Neural Information Processing Systems*, 34:21453–21466, 2021.

[25] M. Bettini, A. Shankar, and A. Prorok. Heterogeneous multi-robot reinforcement learning. *arXiv*, 2023.

[26] J. Zhang, J. Ge, S. Li, S. Li, and L. Li. A bi-level network-wide cooperative driving approach including deep reinforcement learning-based routing. *IEEE Transactions on Intelligent Vehicles*, 2023.

[27] W. Wang, L. Mao, R. Wang, and B.-C. Min. Multi-robot cooperative socially-aware navigation using multi-agent reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2024.

[28] P. Feng, J. Liang, S. Wang, X. Yu, X. Ji, Y. Chen, K. Zhang, R. Shi, and W. Wu. Hierarchical consensus-based multi-agent reinforcement learning for multi-robot cooperation tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

[29] B. Bischoff, D. Nguyen-Tuong, I. Lee, F. Streichert, A. Knoll, et al. Hierarchical reinforcement learning for robot navigation. In *Proceedings of The European Symposium on Artificial Neural Networks, Computational Intelligence And Machine Learning*, 2013.

[30] Y. Jin, S. Wei, J. Yuan, and X. Zhang. Hierarchical and stable multiagent reinforcement learning for cooperative navigation control. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1):90–103, 2021.

[31] J. Wöhlke, F. Schmitt, and H. van Hoof. Hierarchies of planning and reinforcement learning for robot navigation. In *IEEE International Conference on Robotics and Automation*, 2021.

[32] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12):14413–14423, 2020.

[33] W. Zhu and M. Hayashibe. A hierarchical deep reinforcement learning framework with high efficiency and generalization for fast and safe navigation. *IEEE Transactions on Industrial Electronics*, 70(5):4962–4971, 2022.

[34] T. Wu, K. Xue, and P. Wang. Leader-follower formation control of usvs using APF-based adaptive fuzzy logic nonsingular terminal sliding mode control method. *Journal of Mechanical Science and Technology*, 36(4):2007–2018, 2022.

[35] H. Xiao and C. P. Chen. Leader-follower consensus multi-robot formation control using neurodynamic-optimization-based nonlinear model predictive control. *IEEE Access*, 7:43581–43590, 2019.

[36] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya. Multi-robot virtual structure switching and formation changing strategy in an unknown occluded environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[37] N. Abujabal, R. Fareh, S. Sinan, M. Baziyad, and M. Bettayeb. A comprehensive review of the latest path planning developments for multi-robot formation systems. *Robotica*, 41(7): 2079–2104, 2023.

[38] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya. Virtual region based multi-robot path planning in an unknown occluded environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[39] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*, 43:1079–1100, 2019.

[40] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan. Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. *IEEE Robotics and Automation Letters*, 7(3):5896–5903, 2022.

[41] N. Hacene and B. Mendil. Behavior-based autonomous navigation and formation control of mobile robots in unknown cluttered dynamic environments with dynamic target tracking. *International Journal of Automation and Computing*, 18(5):766–786, 2021.

[42] R. Han, S. Chen, and Q. Hao. Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2020.

[43] Q. Li, F. Gama, A. Ribeiro, and A. Prorok. Graph neural networks for decentralized multi-robot path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[44] Z. Gao, G. Yang, and A. Prorok. Co-optimization of environment and policies for decentralized multi-agent navigation. *arXiv*, 2024.

[45] P. Gao, Y. Shen, and M. C. Lin. Collaborative decision-making using spatiotemporal graphs in connected autonomy. *IEEE International Conference on Robotics and Automation*, 2024.

[46] Ö. Özkahraman and P. Ögren. Collaborative navigation-aware coverage in feature-poor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[47] W. Luo, S. Yi, and K. Sycara. Behavior mixing with minimum global and subgroup connectivity maintenance for large-scale multi-robot systems. In *IEEE International Conference on Robotics and Automation*, 2020.

[48] D. Roy, M. Maitra, and S. Bhattacharya. Exploration of multiple unknown areas by swarm of robots utilizing virtual-region-based splitting and merging technique. *IEEE Transactions on Automation Science and Engineering*, 19(4):3459–3470, 2021.

[49] S. Swaminathan, M. Phillips, and M. Likhachev. Planning for multi-agent teams with leader switching. In *IEEE International Conference on Robotics and Automation*, 2015.

[50] S. Novoth, Q. Zhang, K. Ji, and D. Yu. Distributed formation control for multi-vehicle systems with splitting and merging capability. *IEEE Control Systems Letters*, 5(1):355–360, 2020.

[51] Á. Calvo and J. Capitán. Optimal task allocation for heterogeneous multi-robot teams with battery constraints. In *IEEE International Conference on Robotics and Automation*, 2024.

[52] T. Guo, S. D. Han, and J. Yu. Spatial and temporal splitting heuristics for multi-robot motion planning. In *IEEE International Conference on Robotics and Automation*, 2021.

[53] T. Guo and J. Yu. Efficient heuristics for multi-robot path planning in crowded environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

[54] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *International Conference on Representation Learning*, 2017.

[55] Z. Deng, P. Gao, W. J. Jose, and H. Zhang. Multi-robot collaborative navigation with formation adaptation. *arXiv*, 2024.

[56] C. Gabellieri, A. Palleschi, and L. Pallottino. Force-based formation control of omnidirectional ground vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[57] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li. A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6907–6921, 2021.

[58] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[59] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[60] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[61] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *IEEE International Conference on Robotics and Automation*, 2000.

[62] 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*, 36(9):1000–1021, 2017.

[63] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus. Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus. In *IEEE International Conference on Robotics and Automation*, 2016.

[64] B. Reily, T. Mott, and H. Zhang. Adaptation to team composition changes for heterogeneous multi-robot sensor coverage. In *IEEE International Conference on Robotics and Automation*, 2021.

[65] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer. Cooperative multi-robot object transportation system based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 6(4):6466–6472, 2021.

[66] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv*, 2017.

[67] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[68] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[69] T. D. Barfoot. *State estimation for robotics*. Cambridge University Press, 2024.