FACA: Fair and Agile Multi-Robot Collision Avoidance in Constrained Environments with Dynamic Priorities

Jaskirat Singh, Rohan Chandra
Dept. of Computer Science, University of Virginia
{swd3rm, rohanchandra}@virginia.edu

***Abstract*—Multi-robot systems are increasingly being used for critical applications such as rescuing injured people, delivering food and medicines, and monitoring key areas. These applications usually involve navigating at high speeds through constrained spaces such as small gaps. Navigating such constrained spaces becomes particularly challenging when the space is crowded with multiple heterogeneous agents all of which have urgent priorities. What makes the problem even harder is that during an active response situation, roles and priorities can quickly change on a dime without informing the other agents. In order to complete missions in such environments, robots must not only be safe, but also agile, able to dodge and change course at a moment's notice. In this paper, we propose `FACA`, a fair and agile collision avoidance approach where robots coordinate their tasks by talking to each other via natural language (just as people do). In `FACA`, robots balance safety with agility via a novel artificial potential field algorithm that creates an automatic "roundabout" effect whenever a conflict arises. Our experiments show that `FACA` achieves a improvement in efficiency, completing missions more than $3.5\times$ faster than baselines with a time reduction of over 70% while maintaining robust safety margins.**

I. Introduction

Multi-agent robots are extensively used in surveillance and security systems [1], logistics [2], delivering products [3], medical emergency and healthcare [4], motion and traffic analysis, and various other research purposes [5], [6]. In this work, we focus on the problem of safe, scalable, and efficient heterogeneous multi-robot system navigation in cluttered and constrained environments [7], [8]. We consider a distributed (agents are allowed to communicate with each other), but partially observable setting where agents are allowed to communicate with each other. The agents are heterogeneous– they can be UGVs, robots, etc. They have different priorities which can change at any time. The environment consists of various obstacles and small gaps. This problem arises naturally in many crisis situations and presents many issues [9], [10], [11]. Consider a situation where a surveillance agent, a food delivery agent, and a medical emergency agent are navigating through a post-battle environment for rescue and monitoring. If the environment is congested, we expect the medical emergency agent to be granted the right-of-way whereas the surveillance agent to have the least priority. However, things become complex when these priorities shift without warning [12]. For example, the surveillance agent might have their priority escalated to assist in an emergency evacuation, making their mission more urgent than food delivery agent.

The key challenge is that in cluttered and constrained environments, many heterogenous multi-robots are unable to manuever smoothly and effectively to accommodate the sudden change in the mission objectives [13], [14], [15]. There are three key ingredients required to facilitate safe, efficient, and scalable navigation in these scenarios: (1) agents must coordinate with one another and keep track of the dynamic priorities in real-time and (2) agents must be agile when maneuvering through small gaps and around obstacles, and (3) motion planning needs to be cheap and fast in order to scale as well as support re-planning during shifting priorities/objectives.

Motion planning research in multi-robot systems has made significant advances in open spaces, but struggle with cluttered environments [16], [17], [18]. Conventional local planners often fall short in this regard [19], [20]. For instance, while the standard Artificial Potential Field (APF) method is valued for its simplicity and computational efficiency, but it is notoriously susceptible to the local minima problem, where an agent can become trapped in a concave obstacle configuration having the oscillation, failing to reach its goal. On the other end of the complexity spectrum, planners like Model Predictive Control (MPC) can generate optimal, collision-free trajectories but impose a significant computational burden that makes them ill-suited for the rapid, on-the-move re-planning required in dynamic, congested environment. This dichotomy highlights a critical gap in the state-of-the-art: existing methods are often either too computationally expensive for rapid re-planning or they struggle to produce trajectories that are simultaneously safe and efficient in dense, constrained scenarios. Furthermore, a related body of work in the trajectory and intent forecasting community [21], [22], [23], [24], [25], [26], [27], [28], [29], [30] can take in the past few seconds of history and predict the future trajectories in real-time, however, these methods operate at the trajectory level only. The core research question we ask is: *how does a system of heterogeneous multi-robot systems react quickly to changing priorities and objectives while maneuvering safely through small gaps and around obstacles?*

Main Contributions: To answer this question, we make two contributions.

1) First, we allow agents to communicate with each other via natural language, much like humans do. We leverage recent advances in large language models (LLMs) [31], [32], [33] to facilitate a dialogue wherein agents update each other about the changing priorities and resolve any conflicts. Our LLM-driven negotiating module has several advantages. First, it can reason about highly

complex roles, even ones that are difficult to distinguish. Second, we do not require pre-programming roles apriori or any kind of finetuning or data augmentation as LLMs exhibit good zero-shot reasoning abilities.

2) Our second contribution consists of a lightweight motion planner using articial potential fields (APFs). APF methods produce agile trajectories quickly, but are known to be susceptible to local minima issues. In this work, we seek to exploit the low computational requirements and agility of APFs while at the same time proposing a solution to the local minima problem. More specifically, we formulate new tangential and radial potential fields that create a "roundabout" effect to mitigating local minima.

II. Related Work

A. Model-Based Approaches

Since collision avoidance has been a great topic of interest, several approaches for collision avoidance in mult-robot system have been developed over the years which typically falls under reactive and proactive categories. Most of proactive methods rely on the concept of Velocity Obstacles (VO) introduced by [34] which defines the set of all relative velocities that will lead to a collision between a holonomic robot and moving obstacles at some moment in time. VOs offer a tractable alternative to reasoning over inevitable collision states [35], [36] and have been extended to model reciprocal interactions between agents, enabling decentralized multi-agent navigation [37], [38]. Subsequent variants handle robots with more complex dynamics [39], [40], [41], coordinated formations of multiple robots [42], [43], and uncertainty in obstacle motion [44], [45]. Besides the previously mentioned techniques, that is, Generalized Velocity Obstacles (GVO) was developed by [46] to define what it means to collide with obstacles in the velocity space and enables a kinematically constrained robot to make a collision-free maneuver by selecting an available control outside of the collision-safe space. Then to deal with the challenges that sampling-based VOs, omnidirectional robot collision avoidance algorithm (ORCA) was recommended by [47]. Since the original work of Van Den Berg et al. [47] on holonomic robots, many ORCA-based approaches have been proposed that linearize the VOs or learn such constraints, including approaches for steering differential-drives, car-like robots, and other non-holonomic agents [48], [49], [50], [51], [52].

B. Learning-based Approaches

When dynamic environments are taken into consideration the various Deep Reinforcement Learning (DRL) based approaches are widely used for collision avoidance. Building on PPO [53], early work learned fully decentralized policies directly from raw 2D lidar [54], [55], later exploring better lidar state representations [56], multi-layout indoor training [57], and late-fusing lidar with camera images [58]. To incorporate higher-level interaction cues, researchers feed relative positions/velocities and social signals into the policy [59], [60], [61], model human–robot and human–human interactions via attention and relational graphs [62], [63], encode crowds with GCNs [64], and use decentralized structural-RNNs [65]. Yet these agent-level methods often assume open and free spaces and incorpoprtes interaction with the static obstacles; remedies include adding static maps (with dual models for pedestrian presence for safe navigation in dynamic enviornments) [66], late-fusing lidar with pedestrian forecasts [67], and learning lidar latent dynamics unsupervised learning[68]. Beyond perception, reward shaping has been tackled through knowledge distillation [69] and square warning regions [70], though the former depends on expert data quality and the latter tends to be overly conservative.

C. Multi-Robot Communication through LLMs

In decentralized, cooperative multi-robot environments, communication methods are needed for coordination [71]. Researchers are working to integrate LLMs into multi-robot systems (MRS) to address the unique challenges associated with deploying and coordinating MRS [72], [73] For example, effective communication is essential for the MRS to share knowledge, coordinate tasks,and maintain cohesion in the dynamic environment among individual robots [74]. LLMs can provide a natural language interface for inter-robot communication, allowing robots to exchange high-level information more intuitively and efficiently instead of predefined communication structures and protocols [73]. The LLMs can understand the mission, divide it into sub-tasks, and assign them to individual robots within the team based on their capabilities [75], [76]. The generalization ability across different contexts of LLMs can also allow MRS to adapt to new scenarios without extensive reprogramming, making them highly flexible during the deployment [77], [78].

III. Problem Formulation and Background

Assumptions: We assume a distributed system (agents are allowed to communicate) where agents can observe the environment fully, but can only partially observe other agents' states. We further assume agents are self-interested (they optimize their individual objective as opposed to a joint objective), although they are cooperative in the sense that they share information honestly during communication. We also assume agents are point masses (although they can be easily extended to circles).

We begin with a modified Partially Observable Stochastic Game (POSG) [79], defined by the tuple: $\langle N, \mathcal{S}, \mathcal{U}, \mathcal{T} \rangle$. We assume there are N heterogeneous robots in a confined two-dimensional shared space denoted as $S \in \mathbb{R}^2$ at any given time t. A robot's state comprises of its position, $s_i(t) = \big(x_i(t), y_i(t)\big)$ and turning angle $\theta_i(t) \in \mathcal{U}$, $0 \leq \theta_i \leq 2\pi$. A robot's action at time t is expressed as velocity $v_i(t) \in \mathbb{R}^2$ and $0 \leq v_i(t) \leq \bar{v}_i$, where $\bar{v}_i$ is the maximum velocity. Each robot has a goal $g_i \in \mathcal{S}$. The discrete transition function is given as $s_{i+1}(t) = s_i(t) + v_i(t)\cos(\theta_i(t))$. Initially each robot is assigned a priority $\rho_i(t) \in \mathbb{R}$ that may change based on various situations, resulting in the assignment of a new priority at $t+1$. The objective for each robot is to reach its goal as fast as possible. Thus, each robot i solves the following:

$$\begin{aligned}
\arg\min_{\mathcal{U}} \quad & \text{Time to Goal}_i && && \text{(1a)}\\
\text{s.t.} \quad & \|s_i(t) - s_j(t)\| && \geq \text{safe distance} && \text{(1b)}\\
& \|s_i(t) - \text{obstacle}\| && \geq \text{safe distance} && \text{(1c)}\\
& \text{TTG}_i < \text{TTG}_j && \text{if } \rho_i > \rho_j && \text{(1d)}
\end{aligned}$$

Constraint (1b) operates over all other robots j. Constraint (1c) operates over all obstacles. All constraints hold for some fixed horizon in the future. Constraint (1d) operates over all pairs of agents i, j and measures fairness where TTG_i represents the time to goal for robot i. A fair outcome is one where, in cases of conflict, robots with higher priority are expected to be given the right of way, therefore reaching the goal first.

Artificial Potential Fields

We build our algorithm using artificial potential fields [80] which define an attractive potential ($\mathcal{U}_i^A$) towards the goal and a repulsive potential ($\mathcal{U}_i^R$) from the obstacles for collision avoidance. These are then combined to the obtain the net potential ($\mathcal{U}_i^N$) $= \mathcal{U}_i^A + \mathcal{U}_i^R$. The corresponding forces $\mathcal{F}_i^N$, $\mathcal{F}_i^A$, and $\mathcal{F}_i^R$ are then derived by taking the negative gradient of the respective potential functions. The standard artificial potential field implementation uses the following repulsive and attraction potential functions:

$$\mathcal{U}_i^A = \frac{1}{2}\eta^A \|g_i - s_i\|^2, \tag{2}$$

$$\mathcal{U}_i^R(j) = \frac{1}{2}\eta^R \left(\frac{1}{\|s_i - s_j\|} - \frac{1}{\sigma}\right)^2, \tag{3}$$

where $\mathcal{U}_i^R(j)$ is the potential experienced by i due to j and $||\cdot||$ is the euclidean distance between robots i and j or between the robot and the goal, σ is the safety distance, and $\eta^A, \eta^R > 0$, represent the attractive and repulsive gain respectively.

A key limitation of the APF method is that the classical APF method when directly applied to different settings often gets stuck in local minima [80]. Therefore, it needs to be adapted and re-designed for the particular application. Therefore, we introduce a new APF collision avoidance algorithm that mitigates the local minima problem in multi-robot navigation via a smooth roundabout effect.

IV. FACA Framework

Here, we describe FACA in detail. First, we review the technical setup of our environment and provide an overview of our approach. Next, we present our implementation of LLM-based priority determination and the agent-to-agent dialogue negotiation mechanism for task prioritization. Finally, we explain our collision-avoidance strategies for robots, including navigation around obstacle and through confined spaces such as small gaps.

A. Technical Approach

The overview of our proposed technical approach is shown in Figure 1. At a high level, at each discrete time step t, each robot at $s_i(t)$ updates its position to $s_i(t+1)$ by computing a new velocity $v_i(t+1)$ while solving the optimization problem in (1). The optimization proceeds as follows: every robot observe the environment and checks for potential collisions via Equation (4). If a collision is detected, then two things happen. First, the robots engage in an LLM-based dialogue (Section IV-B1) to exchange mission urgency and context information, dynamically adjusting their priorities. Simultaneously, the robots execute the newly proposed “Roundabout Effect” collision avoidance algorithm (Section 2 and 3), which in turn uses the latest priorities, guiding robots to yield or proceed based on negotiated right-of-way. The collision avoidance routine determines a new velocity to execute via Equation (14).

B. Collision Avoidance Around Dynamic Agents: The “Roundabout Effect” (Solving Constraint 1b)

We divided our approach into two parts to avoid collision with other robots. First, we predict the collision by considering a 2D engagement scenario with two robots, i, and j. Let the initial position of i be $s_i(t) = (x_i, y_i)$, and that of j be $s_j(t) = (x_j, y_j)$. The robots have velocities $v_i(t)$, and $v_j(t)$, respectively. We calculate the distance of closest approach, $\Sigma(t)$ as:

$$\begin{aligned}
\Sigma(t)^2 = &\big(s_i(t) - s_j(t)\big)\cdot\big(s_i(t) - s_j(t)\big)\\
&+2\big(s_i(t) - s_j(t)\big)\cdot\big(v_i(t) - v_j(t)\big)t_{col}\\
&+\big(v_i(t) - v_j(t)\big)\cdot\big(v_i(t) - v_j(t)\big)t_{col}^2
\end{aligned} \tag{4}$$

Next, If $\Sigma(t)$ is less than a safe distance threshold, that means it will (1) trigger the LLM-based dialogue communication for priority negotiation as explained in IV-B1 and (2) perform collision avoidance maneuver as described in Sections 2 and IV-B3:

1) LLMs for Priority Determination and Negotiation (Solving Constraint 1d): The main goal of LLM-based dialogue is to satisfy Constraint (1d) by ensuring that higher priority robots are given the right of way in case of a potential collision. Each agent has access to an LLM (gpt-4o-mini [81]) instance through the OpenAI API to perform context-aware, adaptive priority negotiation in real-time, and has an initial prompt which we provide in supplementary material. When the agents observe each other, they take turns sending messages; they receive the last message from the other agent, add it to the dialogue history, and query the LLM for a reply to send back. The robots keep moving around the environment while communicating and will stop conversing if they reach a consensus on which agent’s task has the higher priority. If the consensus leads to a change in the priorities, the new priorities are updated. Additionally, we considered complex scenarios where negotiating robots perform critical operations with similar priorities. As shown in Figure IV.1, a representative case involves *robot i* transporting a patient to a

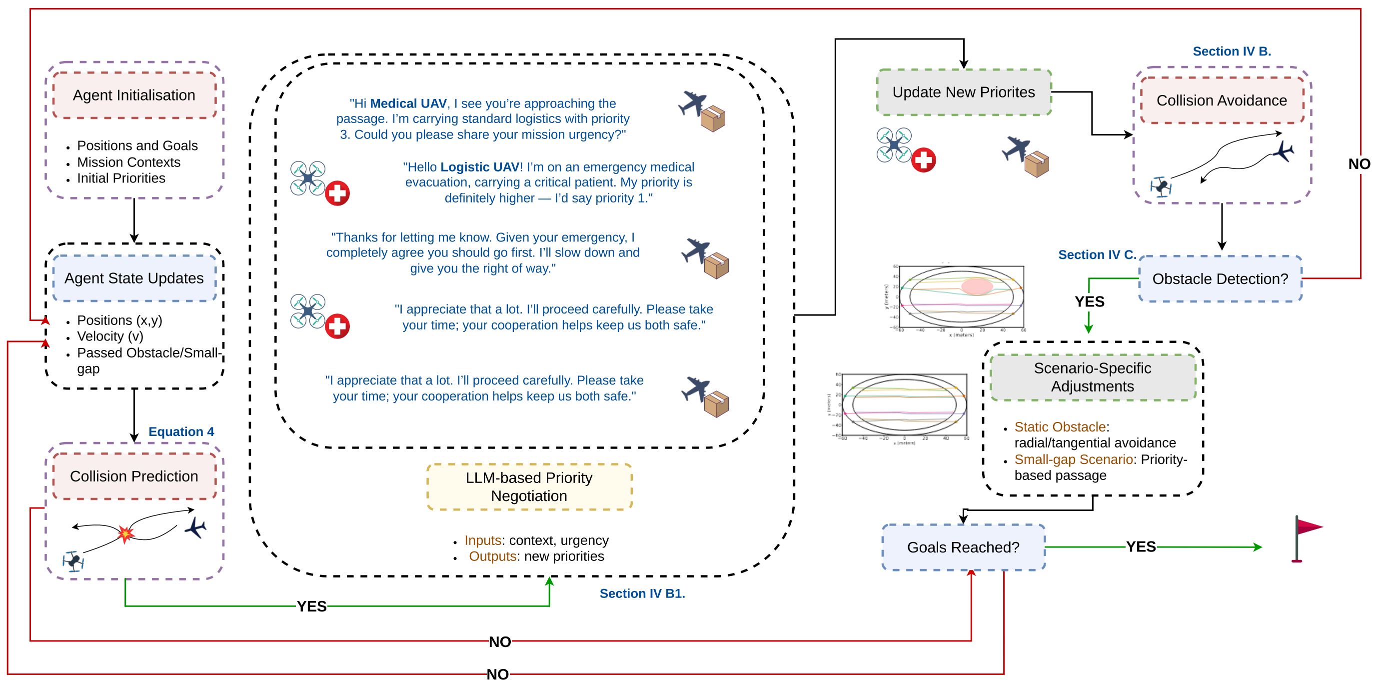


Fig. 1: An overview of the proposed framework for autonomous robot collision avoidance using Large Language Model (LLM) based negotiation in heterogenous robots. The system operates in a loop, beginning with agent initialization. Upon predicting a potential collision, an LLM is used to dynamically negotiate priorities based on mission context. As illustrated in the *LLM-based Priority Negotiation* box where the "Medical robot" is granted higher priority over a "Logistic robot" with inputs as mission, and urgency to output new, dynamically-assigned priorities. The framework then applies scenario-specific adjustments for challenges like obstacle or small-gap passages before executing the final collision avoidance maneuvers.

city hospital while *robot j* transports a ventilator to the same hospital. With both missions assigned the highest priority, the robots negotiate to reach a consensus.

Prompt IV.1: Example LLM-Mediated Dialogue

robot i**:** robot j, conflict warning. I am transporting a critical patient to City General Hospital, my mission is top priority. I must maintain my current heading.

robot j**:** Acknowledged, robot i. I am also on a critical mission, transporting time-sensitive medical equipment for emergency surgery. My priority is also maximal.

robot i**:** Understood. My patient is unstable, and my distance to the hospital is 3.1km. What is your distance to goal?

robot j**:** My distance to City General is 5.8km. Your patient's immediate proximity is more critical. I will yield. I am altering my course and reducing speed. Proceed safely.

Agreement (LLM Output): `"{`i`: high priority, `j`: low priority}"`

2) Non-linear Exponential Attractive Force: One of the issues with the attractive field given in Equation (2) is that the attraction force strength reduces the closer a robot gets to its goal. If there is an obstacle near the goal, the repulsive field from the obstacle overpowers the attractive field and it results in a local minima in the potential that is away from the goal. Therefore, we proposed the following attractive force. We define the potential function $\mathcal{U}_A$ as:

$$\mathcal{U}_A(d) = \kappa^A d - \kappa^A \frac{\sqrt{\pi}}{2\sqrt{-\varphi^A}} \operatorname{erf}(\sqrt{-\varphi^A} d) \tag{5}$$

where d is the distance between the goal and current position of the robot and $\varphi^A < 0$ represents the attractive field spread. Now, $\mathcal{F}^A = -\nabla \mathcal{U}_A(d) = -\left(\frac{\partial \mathcal{U}_A}{\partial d}\right) \widehat{d}$ where $\widehat{d} = \frac{s_i - g_i}{\|s_i - g_i\|}$. Using the standard derivative $\frac{\partial}{\partial z}\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} e^{-z^2}$:

$$\begin{aligned}
\frac{\partial \mathcal{U}_A}{\partial d} &= \frac{\partial}{\partial d}\left[\kappa^A d - \kappa^A \frac{\sqrt{\pi}}{2\sqrt{-\varphi^A}} \operatorname{erf}(\sqrt{-\varphi^A} d)\right] \\
&= \kappa^A - \kappa^A \frac{\sqrt{\pi}}{2\sqrt{-\varphi^A}} \cdot \left(\frac{2}{\sqrt{\pi}} e^{-(\sqrt{-\varphi^A} d)^2}\right) \cdot \sqrt{-\varphi^A} \\
&= \kappa^A - \kappa^A e^{-(-\varphi^A d^2)} \\
&= \kappa^A \left(1 - e^{\varphi^A d^2}\right) \\
&= \kappa^A \left(1 - \exp\left(\varphi^A \|g_i - s_i\|^2\right)\right)
\end{aligned}$$

This term represents the magnitude of the attractive force. Then,

$$\begin{aligned}\mathcal{F}^A &= -\left(\frac{\partial \mathcal{U}_A}{\partial d}\right)\widehat{d} \\ &= -\kappa^A\Big(1-\exp\big(\varphi^A\|s_i-g_i\|^2\big)\Big) \\ &\quad\times\left[\frac{s_i-g_i}{\|s_i-g_i\|}\right]\end{aligned} \tag{6}$$

where $\kappa^A > 0$ represents the attractive gain, which is used to adjust the intensity of the attractive field.

3) Tangential Exponential Repulsive Force: Let $d_{ij} = \|s_i - s_j\|$ denote the distance between robot i and robot j, and let $\widehat{d}_{ij} = \dfrac{s_i - s_j}{d_{ij}}$ denote the corresponding unit vector. We define the repulsive potential as

$$\mathcal{U}_R(d_{ij}) = -\kappa^R \frac{\sqrt{\pi}}{2\sqrt{\varphi^R}}\ \mathrm{erf}\big(\sqrt{\varphi^R}\, d_{ij}\big), \tag{7}$$

where $\kappa^R > 0$ represents the repulsive gain, which is used to adjust the intensity of the repulsive field, and $\varphi^R > 0$ represents the repulsive field spread. Now, $\mathcal{F}_i^R = -\nabla\mathcal{U}_R(d_{ij}) = -\left(\frac{\partial \mathcal{U}_R}{\partial d_{ij}}\right)\widehat{d}_{ij}$. Using the standard derivative $\frac{\partial}{\partial z}\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}e^{-z^2}$:

$$\begin{aligned}\frac{\partial \mathcal{U}_R}{\partial d_{ij}} &= \frac{\partial}{\partial d_{ij}}\left[-\kappa^R \frac{\sqrt{\pi}}{2\sqrt{\varphi^R}}\ \mathrm{erf}\big(\sqrt{\varphi^R}\, d_{ij}\big)\right] \\ &= -\kappa^R\frac{\sqrt{\pi}}{2\sqrt{\varphi^R}}\cdot\left(\frac{2}{\sqrt{\pi}}e^{-(\sqrt{\varphi^R}d_{ij})^2}\right)\cdot\sqrt{\varphi^R} \\ &= -\kappa^R e^{-\varphi^R d_{ij}^2}.\end{aligned}$$

This term represents the slope of the repulsive potential. Then,

$$\begin{aligned}\mathcal{F}_i^R &= -\left(\frac{\partial \mathcal{U}_R}{\partial d_{ij}}\right)\widehat{d}_{ij} \\ &= -\left(-\kappa^R e^{-\varphi^R d_{ij}^2}\right)\widehat{d}_{ij} \\ &= \kappa^R e^{-\varphi^R\|s_i-s_j\|^2}\left[\frac{s_i-s_j}{\|s_i-s_j\|}\right].\end{aligned} \tag{8}$$

To prevent head-on standstills and induce a roundabout effect, the repulsive force is rotated by ninety degrees using the rotation operator $\xi = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. The resulting tangential exponential repulsive force is

$$\begin{aligned}\mathcal{F}_i^R &= \xi\,\mathcal{F}_i^{R,\mathrm{rad}} \\ &= \kappa^R\exp\big(-\varphi^R\|s_i-s_j\|^2\big)\,\xi\left[\frac{s_i-s_j}{\|s_i-s_j\|}\right].\end{aligned} \tag{9}$$

To incorporate priority obtained from the LLM-based dialog discussed in Section IV-B1 into the repulsive force, a scaling factor $\frac{\rho_i}{\rho_i}$ is used. When two robots i and j possess equal priorities, their interaction resembles vehicles navigating a traffic circle, continuously maneuvering around each other to prevent collisions. However, if i is assigned a higher priority mission relative to j (i.e., $\rho_i > \rho_j$), the repulsive force exerted on j by i will be greater compared to the force experienced by i from j. Consequently, j will undergo a greater deviation from its path than i, allowing i to maintain a more direct trajectory towards its objective. The resulting repulsion force is the modified as:

$$\mathcal{F}_i^R = \frac{\rho_i}{\rho_j}\kappa^R\exp\big(-\varphi^R\,\|s_i-s_j\|^2\big)\,\xi\left[\frac{s_i-s_j}{\|s_i-s_j\|}\right] \tag{10}$$

The rotation leads to the creation of the new heading angle $\theta_i(t+1)$ which is simply the angle of this vector as described below:

$$\theta_i(t+1) = \tan^{-1}\left(\mathcal{F}_i^A + \mathcal{F}_i^R\right) \tag{11}$$

C. Collision Avoidance Around Static Obstacles (Solving Constraint 1c)

Our approach to collision avoidance for static obstacles mirrors that of Section IV-B with a key difference. Namely, assuming a circular obstacle with radius r and center μ, the repulsive force $\mathcal{F}_i^R$ is no longer perpendicular to the direction of the robot towards the obstacle. Instead, $\mathcal{F}_i^R$, is less conservative, and is along the tangent direction to the boundary of the obstacle. Formally, let ϕ be defined as the rotation matrix required to turn $s_i - \mu$ so that it becomes the tangent to the obstacle via simple geometry. Then,

$$\mathcal{F}_o^R = \kappa^R\Big(\exp\big(-\varphi^R\,\|s_i-\mu-r\|^2\big)\Big)\phi \tag{12}$$

The term $\exp\big(-\varphi^R\,\|s_i-\mu-r\|^2\big)$ denotes the strength of the force depending on how far the robot is from the nearest point on the boundary of the obstacle. Then, the turning angle can be computed as,

$$\theta_i(t+1) = \tan^{-1}\left(\mathcal{F}_i^A + \mathcal{F}_o^R\right) \tag{13}$$

D. Final Velocity Computation using θ_i

Once the turning angle is determined, it rotates the current velocity vector towards the selected avoidance direction by angle θ to obtain a new velocity $v_i(t+1)$ as shown below.

$$v_i(t+1) = \begin{bmatrix}\cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i\end{bmatrix} v_i(t) \tag{14}$$

To ensure smoothness, we blend the robot's current velocity and the avoidance direction, which is modified by adjusting by calculated turning angle. This blending is done using a factor that increases as the robot gets closer to the obstacle, to smoothly transition from goal-directed movement to avoidance behavior. In order to perform a progressive change in behavior as the robot gets closer to the obstacle, the blending factor (β) is determined which is dependent on the robot's distance from the obstacle boundary. The equation representing this mechanism has been described below:

$$v_i(t+1) \leftarrow \beta v(t) + (1-\beta) v_i(t+1) \tag{15}$$

Remark: To maneuver through small gaps, we use the concept of subgoals. When a robot i starts initially from their respective positions, their motion is attracted toward the center of the gap which acts as subgoal which we denote as ω. The attractive acceleration towards this subgoal is $\Phi_i^{\text{attr}} = \kappa^A \left(1 - e^{-c\|\omega - s_i\|^2}\right) \frac{s_i - \omega}{\|s_i - \omega\|}$ where c controls the rate of exponential decay. Once i, crosses the wall, it changes its goal from the small gap to its final goal g_i. The repulsion forces are the same as used in Section IV-B3.

V. Experiments and Results

We address the following questions:

- *Q1. What are the benefits of the LLM when negotiating consensus between multiple heterogeneous robots?*
- *Q2. How does the proposed collision avoidance framework compare to traditional APF and MPC-based baselines?*
- *Q3. How challenging are constraints? Why should we care about constraints?*

A. Simulation Environment and Scenario Setup

The simulation is setup where $n = 4$ and $n = 8$ robots are initialized with starting and goal positions arranged in a symmetrical pattern on a $\frac{\pi}{16}$ radian arc on a circle with 50 meters radius. Each robot must travel to its goal placed antipodal with respect to its starting position on the arena boundary. The priority for each robot was drawn from a normal distribution with $\mu = 3$ and $\sigma = 1$. Initially assigned priorities gets dynamically changed to another set of priorities in the middle of a simulation. We generate and add two different sources of complexity. The first spatial complexity consists of a circular shaped obstacle with a fixed radius placed in the arena for creating a more dense and complex environment. The second complexity consists of a small gap between two vertical walls placed at $x = 0$. Simulations were performed in Python in 8-core AMD Ryzen™ 9 5900HS CPU and GeForce RTX™ 3060 GPU. Parameter values used in simulation are provided in the supplementary material. Each scenario is averaged across 100 simulations with a different random seeds.

B. Evaluation Criteria and Baselines

First, we measure the Time to Goal (TTG) to evaluate the overall time taken by the robots to reach their goal. Next, we would like to measure how much distance each robot maintains form other robots, the obstacle, and the walls. That is, the robots should not aim to shoot towards their destination at the risk of colliding or cutting closer to other objects. So we also measure the Mean Minimum Distance (MMD) between all the agents. Next, we measure the Flow Rate (FR) which measures the flow through a gap where multiple agents are trying to get through. It is defined as: $FR = \frac{N}{zT}$ where N is the number of agents, T is the makespan (in seconds), while z is the gap width (in unit meters). Flow rate gives a metric quantifying the overall "volume of agents" successfully navigating a constrained space.

Scenarios	TTG (s)		MMD (m)		FR	
	$n=4$	$n=8$	$n=4$	$n=8$	$n=4$	$n=8$
No LLM, No Constraints	8.79	10.02	1.98	1.49	-	-
With LLM, No Constraints	2.51	2.86	1.95	1.47	-	-
With LLM, obstacle only	8.24	9.63	1.70	1.33	-	-
With LLM, Small Gap only	2.96	3.00	0.17	0.21	2.70	5.00

TABLE I: Evaluating `FACA` in cases with varying spatial constraints with 4 and 8 agents. Performance is evaluated based on the Time-to-Goal (TTG) in seconds and the Mean Minimum Distance (MMD) in meters. FR is measured at small gaps, hence only shown for *row 4*.

Methods	TTG		MMD		FR	
	$n=4$	$n=8$	$n=4$	$n=8$	$n=4$	$n=8$
Classical APF w/o LLM	12.50	12.50	0.27	0.28	0.60	1.20
Classical APF	9.80	12.43	0.19	0.24	0.80	1.20
MPC w/o LLM	4.76	7.78	0.29	0.23	2.10	3.90
MPC	7.92	10.51	0.20	0.21	1.50	4.10
`FACA` w/o LLM	2.95	**3.00**	0.17	0.22	2.60	**5.10**
`FACA`	**2.88**	3.13	0.18	0.25	**2.70**	4.90

TABLE II: Comparing `FACA` with standard collision avoidance baselines. **Bold** is best.

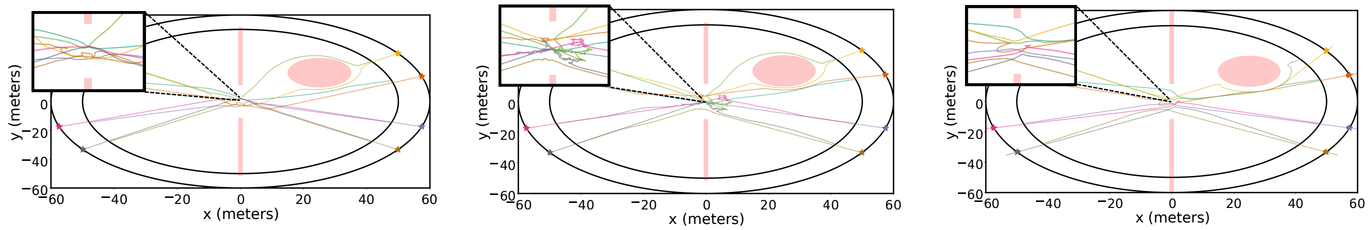
Our baselines consists of the classical APF method utilizing the attraction and repulsion forces given by Equations (2) and (3), respectively, and an MPC baseline, to demonstrate the collision avoidance and navigation capabilities of our method. We also conduct an ablation study where we probe each component of our approach in isolation. Finally, we also compare our approach with an existing state-of-the-art multi-robot method, RIPNA [82], which implements multi-robot navigation but without priority handling.

C. The Challenge of Spatial Constraints

In Table I, we demonstrate the challenging nature of cluttered and constrained environments. Specifically, by removing all spatial constraints (Table I, *row 2*), the robots are the fastest in reaching their goal, almost $3\times$ faster than in the presence of an obstacle. Additionally the robots respect a healthy distance from each other, resulting in a high MMD. On the other hand, the small gap constraint slowed the robots, but not by much. However, the robots cut too close to each other when navigating through a small gap, which is why the MMD reduces significantly.

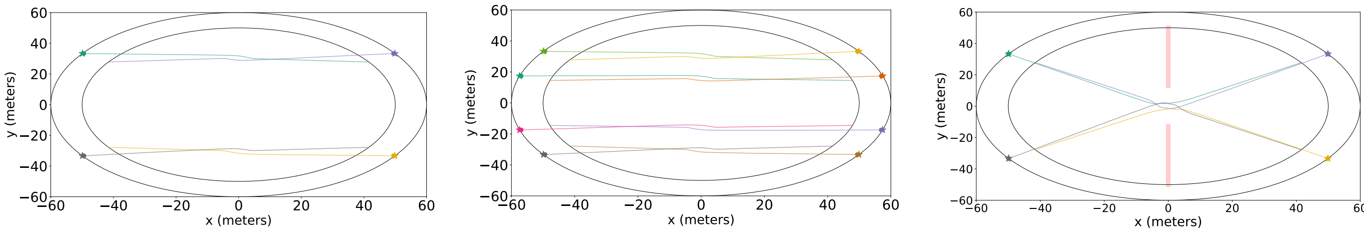
D. The Role and Importance of the LLM

We measured whether the final order in which agents reached their goals matched the final priority order after the priorities were updated during navigation. Across 100 simulations, the LLM-enabled agents managed to reach their goals in the correct order 100% of the time. This was despite the priorities changing multiple times, often drastically–an agent with the lowest priority in the beginning would be later assigned the highest priority, and it would be able to catch up and arrive first to the goal. On the other hand, as expected,

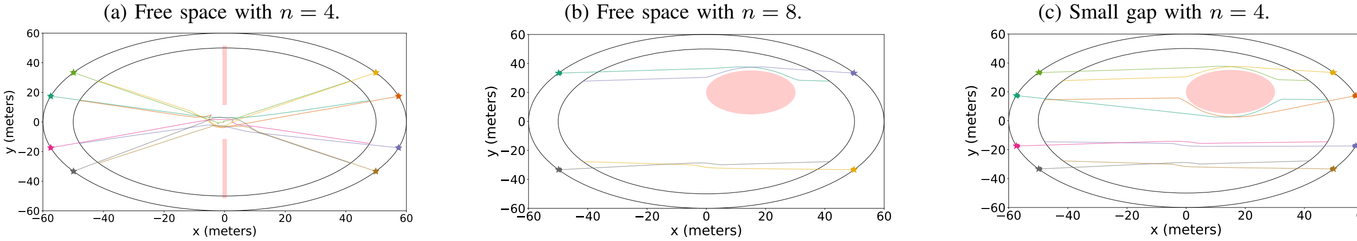


(a) FACA (b) Classical APF (c) MPC

Fig. 2: Comparing FACA with baseline collision avoidance methods.



(a) Free space with $n = 4$. (b) Free space with $n = 8$. (c) Small gap with $n = 4$.



(d) Small gap with $N = 8$. (e) Static obstacle with $N = 4$. (f) Static obstacle with $N = 8$.

Fig. 3: Visually demonstrating the Roundabout effect of FACA in both complex and simple scenarios.

without the LLM, the final ordering of the agents did not match the final priority order even once. Furthermore, when testing the role of LLM without any spatial constraints, the benefits are even more pronounced. Not only does the LLM respect the priority ordering, but it also results in much faster navigation ($> 4\times$ speedup, Table I, *rows 1 and 2*).

E. The Significance of the Roundabout Effect

The roundabout effect balances safety and efficiency. In Table I, we see that FACA is the fastest, often by orders of more than $3\times$ (and thereby maximizing efficiency), without being both overly conservative as well as overly aggressive. This can be observed from the MMD values of FACA which are neither to low nor too high - indicating that FACA can balance the right amount of space to give other robots while at the same time making quick progress towards the goal. Furthermore, the roundabout effect results in the highest flow rate for FACA compared to other baselines. We demonstrate the roundabout effect in Figure 3.

F. Comparing FACA with classical APF, MPC, and RIPNA

Figure 2 and Table II compare FACA with classical APF visually and quantitatively, respectively. We observe that FACA achieves a significantly lower TTG. For instance, with LLM-based dialogue and $n = 8$, FACA is approximately $4\times$ faster than classical APF and $3.4\times$ faster than MPC. This performance advantage is consistent even in the absence of an LLM, where FACA is over $4\times$ faster than classical APF and $2.6\times$ faster than MPC, while also achieving a $4.25\times$ higher FR than classical APF. Finally, we also compared FACA with RIPNA. We observed that FACA improved the TTG by 54.68% and MMD by 58.63%.

Classical APF takes extremely long to reach their goal (high TTG) correspondingly results in a poor FR (Table II, *rows 1 and 2*). The large TTG can also be correlated with the high MMD which implies that the agents are very conservative and try to steer away from other robots as much as possible, which costs time. This can also be visually confirmed in Figure 2b where the agents enter a traffic jam at the gap. MPC performs similarly poorly and repels agents and pushes them back (Figure 2c). MPC also results in lower TTG and highest MMD on account of higher conservativeness (Table II, *rows 3 and 4*). In Figure 2, we additionally note that MPC is reactive, cutting too close to the obstacle as opposed to the obstacle, which is suboptimal. In contrast, the roundabout effect created by FACA results in a balance between safety and agility resulting in lower TTG and high FR.

VI. Conclusion

We presented FACA, a new approach to priority-based collision avoidance within heterogeneous robots through

LLMs in dense, constrained, and crowded airspace that contains obstacles and small-gap passages. The robots communicate with each other by querying the LLMs to generate messages and engage in the dialogue conversation in order to know each other priorities and define new priorities based upon the operations. We demonstrate our approach's effectiveness in simulated constrained environments and compared the results with other baseline and existing algorithms. `FACA` can be further extended to include dynamics of the vehicle, extend in 3D and also include hard constraints like control barrier function to ensure no vehicle enter the safe of other vehicles.

References

[1] S. Witwicki, J. C. Castillo, J. Messias, J. Capitan, F. S. Melo, P. U. Lima, and M. Veloso, "Autonomous surveillance robots: A decision-making framework for networked muiltiagent systems," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 52–64, 2017.

[2] J. Fischer, C. Lieberoth-Leden, J. Fottner, and B. Vogel-Heuser, "Design, application, and evaluation of a multiagent system in the logistics domain," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1283–1296, 2020.

[3] B. Li and H. Ma, "Double-deck multi-agent pickup and delivery: Multi-robot rearrangement in large-scale warehouses," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3701–3708, 2023.

[4] F. Lanza, V. Seidita, and A. Chella, "Agents and robots for collaborating and supporting physicians in healthcare scenarios," *Journal of biomedical informatics*, vol. 108, p. 103483, 2020.

[5] E. T. S. Alotaibi and H. Al-Rawi, "Multi-robot path-planning problem for a heavy traffic control application: A survey," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, 2016.

[6] S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, L. Ravi, and V. Subramaniyaswamy, "Unmanned aerial vehicle (uav) based forest fire detection and monitoring for reducing false alarms in forest-fires," *Computer Communications*, vol. 149, pp. 1–16, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419308655

[7] R. Chandra, S. Singh, W. Luo, and K. Sycara, "Multi-robot navigation in social mini-games: Definitions, taxonomy, and algorithms," *arXiv preprint arXiv:2508.13459*, 2025.

[8] R. Chandra, V. Zinage, E. Bakolas, P. Stone, and J. Biswas, "Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions," *Autonomous Robots*, vol. 49, no. 2, p. 12, 2025.

[9] V. Mahadevan, S. Zhang, and R. Chandra, "Gamechat: Multi-llm dialogue for safe, agile, and socially optimal multi-agent navigation in constrained environments," *arXiv preprint arXiv:2503.12333*, 2025.

[10] R. Chandra and D. Manocha, "Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2676–2683, 2022.

[11] R. Chandra, M. Wang, M. Schwager, and D. Manocha, "Game-theoretic planning for autonomous driving among risk-aware human drivers," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2876–2883.

[12] R. Chandra, R. Maligi, A. Anantula, and J. Biswas, "Socialmapf: Optimal and efficient multi-agent path finding with strategic agents for social navigation," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3214–3221, 2023.

[13] V. Zinage, A. Jha, R. Chandra, and E. Bakolas, "Decentralized safe and scalable multi-agent control under limited actuation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 12 105–12 111.

[14] J. Chen and R. Chandra, "Livepoint: Fully decentralized, safe, deadlock-free multi-robot control in cluttered environments with high-dimensional inputs," *arXiv preprint arXiv:2503.13098*, 2025.

[15] S. Gouru, S. Lakkoju, and R. Chandra, "Livenet: Robust, minimally invasive multi-robot control for safe and live navigation in constrained environments," *arXiv preprint arXiv:2412.04659*, 2024.

[16] N. Suriyarachchi, R. Chandra, J. S. Baras, and D. Manocha, "Gameopt: Optimal real-time multi-agent planning and control at dynamic intersections," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE doi - 10.1109/ITSC55140.2022.9921968, 2022, pp. 2599–2606.

[17] N. Suriyarachchi, R. Chandra, A. Anantula, J. S. Baras, and D. Manocha, "Gameopt+: Improving fuel efficiency in unregulated heterogeneous traffic intersections via optimal multi-agent cooperative control," *arXiv preprint arXiv:2405.16430*, 2024.

[18] R. Chandra, Z. Sprague, and J. Biswas, "Socialgym 2.0: simulator for multi-robot learning and navigation in shared human spaces," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 778–23 780.

[19] A. H. Raj, Z. Hu, H. Karnan, R. Chandra, A. Payandeh, L. Mao, P. Stone, J. Biswas, and X. Xiao, "Rethinking social robot navigation: Leveraging the best of two worlds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 330–16 337.

[20] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra *et al.*, "Principles and guidelines for evaluating social robot navigation algorithms," *ACM Transactions on Human-Robot Interaction*, vol. 14, no. 2, pp. 1–65, 2025.

[21] X. Wu, R. Chandra, T. Guan, A. Bedi, and D. Manocha, "Intent-aware planning in heterogeneous traffic via distributed multi-agent reinforcement learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 446–477.

[22] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8483–8492.

[23] R. Chandra, T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera, and D. Manocha, "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4882–4890, 2020.

[24] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha, "Robusttp: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs," in *Proceedings of the 3rd ACM Computer Science in Cars Symposium*, 2019, pp. 1–9.

[25] R. Chandra, U. Bhattacharya, T. Mittal, A. Bera, and D. Manocha, "Cmetric: A driving behavior measure using centrality functions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2035–2042.

[26] R. Chandra, U. Bhattacharya, T. Mittal, X. Li, A. Bera, and D. Manocha, "Graphrqi: Classifying driver behaviors using graph spectrums," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4350–4357.

[27] R. Chandra, "Towards autonomous driving in dense, heterogeneous, and unstructured traffic," Ph.D. dissertation, University of Maryland, College Park, 2022.

[28] R. Chandra, A. Bera, and D. Manocha, "Using graph-theoretic machine learning to predict human driver behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2572–2585, 2021.

[29] R. Chandra, U. Bhattacharya, T. Randhavane, A. Bera, and D. Manocha, "Roadtrack: Realtime tracking of road agents in dense and heterogeneous environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1270–1277.

[30] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Densepeds: Pedestrian tracking in dense crowds using front-rvo and sparse features," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 468–475.

[31] K. Song, R. Jiang, R. Chandra, and S. Zhang, "Group fairness in multi-task reinforcement learning," *arXiv preprint arXiv:2503.07817*, 2025.

[32] K. Song, A. Moeini, P. Wang, L. Gong, R. Chandra, Y. Qi, and S. Zhang, "Reward is enough: Llms are in-context reinforcement learners," *arXiv preprint arXiv:2506.06303*, 2025.

[33] A. Moeini, J. Wang, J. Beck, E. Blaser, S. Whiteson, R. Chandra, and S. Zhang, "A survey of in-context reinforcement learning," *arXiv preprint arXiv:2502.07978*, 2025.

[34] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, p. 760 to 772, 1998.

[35] T. Fraichard and H. Asama, "Inevitable collision states a step towards safer robots," *Advanced Robotics*, vol. 18, no. 10, p. 1001 to 1024, 2004.

[36] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, p. 2210 to 2215.

[37] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real time multi agent navigation," in *IEEE International Conference on Robotics and Automation*, 2008, p. 1928 to 1935.

[38] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, "Clearpath: highly parallel collision avoidance for multi agent simulation," in *ACM SIGGRAPH Eurographics Symposium on Computer Animation*, 2009, p. 177 to 187.

[39] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration velocity obstacles," in *IEEE International Conference on Robotics and Automation*, 2011, p. 3475 to 3482.

[40] M. Rufli, J. Alonso Mora, and R. Siegwart, "Reciprocal collision avoidance with motion continuity constraints," *IEEE Transactions on Robotics*, vol. 29, no. 4, p. 899 to 912, 2013.

[41] D. Bareiss and J. van den Berg, "Reciprocal collision avoidance for robots with linear dynamics using LQR obstacles," in *IEEE International Conference on Robotics and Automation*, 2013, p. 3847 to 3853.

[42] A. Kimmel, A. Dobson, and K. Bekris, "Maintaining team coherence under the velocity obstacle framework," in *International Conference on Autonomous Agents and Multiagent Systems*, 2012, p. 247 to 256.

[43] I. Karamouzas and S. J. Guy, "Prioritized group navigation with formation velocity obstacles," in *IEEE International Conference on Robotics and Automation*, 2015, p. 5983 to 5989.

[44] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," in *IEEE International Conference on Robotics and Automation*, 2007, p. 1610 to 1616.

[45] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, p. 696 to 706, 2011.

[46] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, p. 5573 to 5578.

[47] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n body collision avoidance," in *Robotics Research The International Symposium ISRR*, ser. Springer Tracts in Advanced Robotics. Springer, 2011, vol. 70, p. 3 to 19.

[48] A. Giese, D. Latypov, and N. M. Amato, "Reciprocally rotating velocity obstacles," in *IEEE International Conference on Robotics and Automation*, 2014.

[49] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Smooth and collision free navigation for multiple robots under differential drive constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, p. 4584 to 4589.

[50] J. Alonso Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non holonomic robots," in *Distributed Autonomous Robotic Systems*. Springer, 2013, p. 203 to 216.

[51] J. Alonso Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car like robots," in *IEEE International Conference on Robotics and Automation*, 2012, p. 360 to 366.

[52] P. Long, W. Liu, and J. Pan, "Deep learned collision avoidance policy for distributed multiagent navigation," *Robotics and Automation Letters*, vol. 2, no. 2, p. 656 to 663, 2017.

[53] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[54] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[55] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[56] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6053–6060.

[57] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1140–1146.

[58] X. Huang, H. Deng, W. Zhang, R. Song, and Y. Li, "Towards multi-modal perception-based navigation: A deep reinforcement learning method," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4986–4993, 2021.

[59] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[60] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[61] ——, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10 357–10 377, 2021.

[62] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[63] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 007–10 013.

[64] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[65] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3517–3524.

[66] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[67] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, "Densecavoid: Real-time navigation in dense crowds using anticipatory behaviors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 345–11 352.

[68] D. Dugas, J. Nieto, R. Siegwart, and J. J. Chung, "Navrep: Unsupervised representations for reinforcement learning of robot navigation in dynamic human environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7829–7835.

[69] P. Xu and I. Karamouzas, "Human-inspired multi-agent navigation using knowledge distillation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8105–8112.

[70] U. Patel, N. K. S. Kumar, A. J. Sathyamoorthy, and D. Manocha, "Dwa-rl: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6057–6063.

[71] R. Chandra, V. Zinage, E. Bakolas, P. Stone, and J. Biswas, "Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions," 2025. [Online]. Available: https://arxiv.org/abs/2308.10966

[72] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?" in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4311–4317.

[73] Z. Mandi, S. Jain, and S. Song, "Roco: Dialectic multi-robot collaboration with large language models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 286–299.

[74] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," *Current robotics reports*, vol. 3, no. 4, pp. 213–225, 2022.

[75] J. Chen, C. Yu, X. Zhou, T. Xu, Y. Mu, M. Hu, W. Shao, Y. Wang, G. Li, and L. Shao, "Emos: Embodiment-aware heterogeneous multi-robot operating system with llm agents," *arXiv preprint arXiv:2410.22662*, 2024.

[76] K. Liu, Z. Tang, D. Wang, Z. Wang, X. Li, and B. Zhao, "Coherent: Collaboration of heterogeneous multi-robot system with large language models," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 208–10 214.

[77] Y. Wang, R. Xiao, J. Y. L. Kasahara, R. Yajima, K. Nagatani, A. Yamashita, and H. Asama, "Dart-llm: Dependency-aware multi-robot task decomposition and execution using large language models," *arXiv preprint arXiv:2411.09022*, 2024.

[78] B. Yu, H. Kasaei, and M. Cao, "Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models," *arXiv preprint arXiv:2310.07937*, 2023.

[79] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI*, vol. 4, 2004, pp. 709–715.

[80] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.

[81] OpenAI, :, A. Hurst, A. Lerer, A. P. Goucher, and et. al., "Gpt-4o system card," 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[82] J. M. George and D. Ghose, "A reactive inverse pn algorithm for collision avoidance among multiple unmanned aerial vehicles," *2009 American Control Conference*, pp. 3890–3895, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:41548167