IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Learning Behaviours for Decentralised Multi-Robot Collision Avoidance in Constrained Pathways Using Curriculum Reinforcement Learning

Md Mostafizur Rahman Komol, *Member, IEEE*, Brendan Tidd, Will Browne, *Member, IEEE*, Frederic Maire, Jason Williams, *Senior Member, IEEE*, and David Howard, *Senior Member, IEEE*

***Abstract*—Mobile robot teams often require decentralised autonomous navigation through narrow gaps in limited communication environments (e.g., underground search-and-rescue operations). Existing navigation approaches exhibit suboptimal performance for avoiding multi-robot collisions in such bottlenecks due to an inability to address the dynamic nature of the robots. Initial work utilising reinforcement learning has demonstrated success in navigating a single robot through narrow gaps. However, when training agents to produce give-way behaviour for navigating through constrained gaps, end-to-end reinforcement learning using simple rewards suffers from slow convergence due to the increased search space of viable policies. This paper introduces a novel curriculum reinforcement learning framework, incorporating a *multi-robot bootstrap curriculum* with preprogrammed behaviour to guide initial policy formation, subsequently refined by a *gap curriculum* that progressively reduces training complexity towards an optimal policy. This framework learns multi-robot interaction behaviours, which are impractical to program manually. Our model achieves a 99% success-rate in give-way behaviour generation without inter-agent communications in high-fidelity simulations. The success-rate reduced to 73% in simulations incorporating noisy sensors, and 60% in field-robot tests, substantiating our model's practical viability despite sensor noise and real-world uncertainties. The simple benchmark methods lack efficiency in basic interaction behaviours.**

***Index Terms*—Multi-robot systems, reinforcement learning, collision avoidance, field robotics, search and rescue robots.**

I. Introduction

COOPERATIVE multi-robot navigation is often required for large-scale mission-critical operations involving a high level of robot autonomy. In search and rescue operations, e.g., in a mining collapse or bushfire, robots are deployed to navigate areas (like underground tunnels or forests) where communication facilities are often compromised [1]. These environments frequently feature narrow gaps, presenting navigational challenges [2]. Coordinating multiple robots approaching these gaps simultaneously[1] can lead to bottlenecks [3], especially when only one robot can pass through at a time. Such situations demand precise timing and coordination. Robots moving toward the gap to maximise goal success may encounter decision-making dilemmas to avoid inter-robot collisions in these bottleneck zones. Conventional autonomous navigation approaches assume static environments. Even when costs are recalculated at each timestep, these approaches overlook the domain's temporal dynamics caused by multiple robots [4], [5], [6], [7]. This oversight can result in suboptimal policies that may lead to collisions or create blocked paths, particularly in environments with noisy sensors. Fig. 1 depicts the problem with scenario trajectories. It includes a real-world example with a potential give-way behaviour solution during autonomous navigation. Hand-engineered programming of behaviours requires extensive knowledge of all potential scenarios and often leads

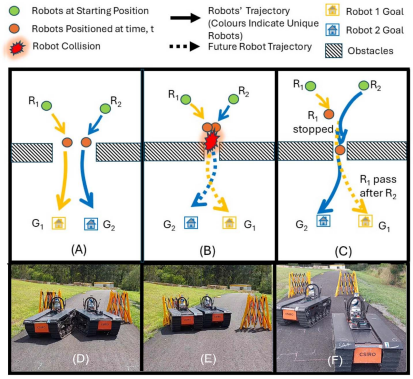

Fig. 1. (A) Wide gap for simultaneous multi-robot passage (B) Narrow gap scenario where give-way behaviour is required (C) Give-way behaviour for avoiding multi-robot collision while passing simultaneously towards a narrow gap (D) A Multi-Robot Collision Scenario with potential risks of damaging robot chassis and sensors (E) Multi-Robot Collision can drag one robot to cause a collision with obstacles (F) Give-way Behaviour preventing multi-robot collision.

Received 21 January 2025; accepted 2 June 2025. Date of publication 19 June 2025; date of current version 15 July 2025. This article was recommended for publication by Associate Editor P. Falco and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported in part by Data61, Robotics and Autonomous Systems Group, the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australian Government, and in part by the Centre for Robotics, Queensland University of Technology (QUT), Australia. *(Corresponding author: Md Mostafizur Rahman Komol.)*

Md Mostafizur Rahman Komol and Will Browne are with Robotics Groups, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Pullenvale, QLD 4069, Australia, and also with the School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, QLD 4000, Australia (e-mail: m.komol@qut.edu.au).

Brendan Tidd, Jason Williams, and David Howard are with the Robotics Groups, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Pullenvale, QLD 4069, Australia.

Frederic Maire is with the School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, QLD 4000, Australia.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3581430, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3581430

[1]In this paper, "simultaneous operation" describes two side-by-side robots approaching a constrained gap at the same time. This inherently presents a potential for trajectory conflicts unless managed by a give-way strategy.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

to suboptimal policies, especially when unanticipated real-world circumstances occur [2].

Alternatively, reinforcement learning has been found effective for a single robot in navigating constrained passageways even when using noisy, local sensors [2]. Multi-agent reinforcement learning has also been applied for multi-robot obstacle avoidance, formation making, and swarm collision avoidance [3], [8], [9]. However, training robots to give way and navigate through optimal trajectories in restricted spaces, requires much interaction with potentially conflicting trajectories. End-to-end reinforcement learning often faces slow convergence when a target task is difficult due to high exploration challenges [10]. Designing complex sparse reward structures incurs substantial tuning costs, causes very slow learning and can often result in a lack of generalisation [10]. Curriculum reinforcement learning offers a progressive learning approach that guides agents from simpler to more complex tasks, enhancing convergence ability through generalising exploration across intermediate guided scenarios [10].

This research aims to coordinate multi-robot interactions by developing a give-way behaviour, wherein one robot will yield, and the other will lead while navigating narrow gaps using a shared reinforcement learning policy. We hypothesise that a reinforcement learning policy, guided by curriculum learning, will effectively address the slow convergence when using a simple reward structure. A *multi-robot bootstrap curriculum* using our reinforcement learning framework is proposed to progressively learn, initially from a sub-optimal, human-biased, hand-coded behaviour. Subsequently, a *gap curriculum* refines the policy from flexible navigation scenarios to learn cooperative behaviours while increasing the difficulty of constrained gap scenarios. Developing our multi-agent reinforcement learning policy for the targeted constrained scenarios requires efficient training due to high computational cost; therefore, we have utilised a lightweight physics simulator PyBullet, leveraging CPU parallelisation to enhance computational efficiency. In addition to PyBullet, our model was tested in the Gazebo simulator using sensors and has demonstrated robustness and generalisability in dynamic, real-world environments.

Our research contributions are as follows:

- This research introduces a novel approach that incorporates a *Multi-robot Bootstrap Curriculum* and a *Gap Curriculum* within a decentralised multi-agent reinforcement learning framework. Our method develops a give-way behaviour policy by progressively learning from a preprogrammed behaviour, and then exploring the target task, while end-to-end learning fails due to slow convergence.
- Our model has been evaluated using two decentralised homogeneous BIA5 robots.[2] They use local sensor maps for simultaneous navigation through a highly constrained gap relative to robots' dimensions (e.g., passing 0.78 m wide robots through a 0.85 m gap). Our model has outperformed the benchmarks: End-to-end reinforcement learning, the Hybrid A* algorithm [11] and two rule-based approaches, which failed to effectively demonstrate multi-robot interaction behaviours.
- Our proposed method has generated emerging behaviours beyond our initially coded primary give-way behaviour. These behaviours were unanticipated prior to training and helped to avoid human bias and inefficiencies in preprogrammed actions. This offers insights into various interactions and scenarios that may occur while navigating constrained environments.

II. Literature Review

A. Traditional Methods

Classical algorithms such as A*, Mixed integer linear programming (MILP), Model predictive control (MPC), Probabilistic road map (PRM), and Rapidly exploring random tree (RRT) are effective for path-finding but face significant challenges in decentralised multi-robot navigation through constrained environments without inter-robot communication [4], [5], [6], [7]. A* variants struggle with scalability in dynamic spaces, whereas PRM and RRT encounter issues with dimensional complexity and inefficient exploration in constrained search spaces [4], [5], [6], [7]. MPC and consensus-based methods rely on communication, making them impractical in local sensor-only environments with limited information sharing [3], [12]. Optimisation and rule-based approaches often fail due to high computational costs, poor generalisation, and inefficient handling of non-linear constraints, which lead to blocked pathways or inefficient overtaking [13]. The artificial potential field method is prone to oscillations near narrow gaps [14] and struggles to accurately predict the movements of dynamic agents, particularly when the dynamic characteristics are not explicitly modelled.

The Hybrid A* algorithm is employed for navigation planning on a graph derived from a height-map or cost-map, dynamically costing edges based on motion primitives [11]. However, Hybrid A* faces challenges like frequent recalculations of edge costs, which lead to extensive search delays. The inability to predict future states can also result in suboptimal policies, as following the greedy action immediately may lead to future trajectory conflicts when robots move independently [1].

In summary, classical methods using local sensor maps, struggle with non-communicating agent dynamics and sensor inaccuracies. They often fail to recognise traversable gaps and require extensive manual tuning, thereby limiting adaptability and scalability in dynamic, constrained environments.

B. Learning Methods

Learning-based methods adapt and scale effectively across various scenarios, but supervised learning requires extensive annotations, unsupervised learning lacks task-specific guidance, and imitation learning struggles with dynamic conditions. For example, the DAGGER algorithm iteratively refines a policy by aggregating training datasets from each iteration, incorporating a mixture of expert-labelled actions in policy-visited states and actions derived directly from the evolving policy [15]. However, this approach heavily relies on expert input, which can be expensive for continuous action problems such as constrained multi-robot coordination [15].

Reinforcement learning has enabled robots to interact with unpredictable environments through real-time decision-making and feedback [16]. In decentralised multi-robot collision avoidance applications, reinforcement learning has been used to navigate, avoiding obstacles and conflicts. Curriculum learning has been employed to incrementally increase training complexity in the reinforcement learning framework by adjusting obstacle density, robot numbers, and goal distances [9], [17]. A shared multi-agent reinforcement learning framework has been applied for training competitive robot soccer teams interacting in a dynamic environment [18]. The Perceptual Hallucination for Hallway Passing (PHHP) approach has been used for two

[2]https://bia5.com/ (Tracked Unmanned Ground Robot of length 1.4 m and width 0.78 m with min-max linear velocity limit [-0.75 m/s,0.75 m/s] and min-max angular velocity limit [-0.75 rd/s, 0.75rd/s])

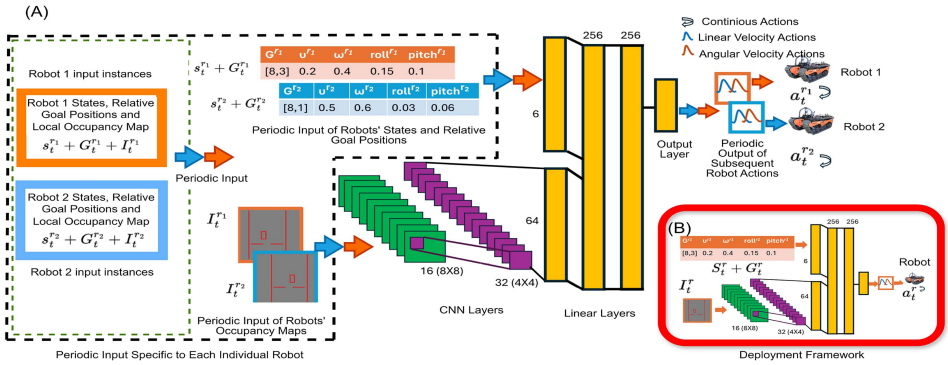

Fig. 2. The shared neural network of the same architecture used for the actor and critic in MAPPO, (A) trained with data collected from all robots with periodic updates ensuring each robot has the latest model and (B) deployed in a decentralised manner for each robot separately. Agents learn cooperative behaviours for navigation by interacting within a constrained environment, rather than following any inter-agent communication protocols. Different colour arrows indicate the periodic input of different robots. Rather than continuously integrating real-time data from the other robot, the policy periodically receives the states and goal positions of two robots (tensor shape [2, 6]) and their corresponding occupancy maps (tensor shape [2, 64]).

opposite-heading robots to pass each other in narrow hallways through the synthetic generation of virtual obstacles, demonstrating improved performance in real-world experiments. But this method is impractical with local sensors in limited communication environments [19]. Curriculum Reinforcement Learning has also been applied for adaptive multi-robot obstacle avoidance and formation-making, utilising spatial-temporal cooperation rather than absolute coordinates to progressively introduce complexity [3]. Despite advances, simultaneous multi-robot coordination in constrained bottlenecks struggles with prioritising give-way behaviour. This study proposes the adoption of curriculum learning strategies to enhance the reward signal and facilitate learning convergence when a simple reward function is used. We hypothesise that our proposed curricula, by guiding the policy toward higher rewards, will reduce the need for hard tuning of rewards, potentially enhancing generalisation and producing emerging cooperative behaviours for multi-robot coordination in congestion scenarios.

III. Method

Our approach involves developing a give-way behaviour to prevent collisions as two decentralised robots approach a bottleneck side by side. We considered a constrained gap that allows only one robot to pass at a time, using local sensor maps without inter-agent communication. The basis of this work is to integrate the behaviour into Hines et al.'s multi-robot autonomous operation stack [11], activating when two robots approach a narrow congestion.

A. Multi-Robot Reinforcement Learning

We view the problem through the frame of a Markov Decision Process (MDP) to effectively model the dynamic and uncertain nature of robots while interacting in constrained environments. We adopt the Proximal Policy Optimisation (PPO) with the clipped surrogate objective for our Multi-Agent PPO (MAPPO) framework [18], [20]. The framework produces one shared policy and value function during training that can be used independently during tests. Robots periodically input their state information into the framework. The centralised critic then evaluates the policy's actions, which are applied to both robots, albeit independently. The robots' input observation o_t includes each robot's state information s_t^r, relative goal position G_t^r (x and y coordinates) and local occupancy map I_t^r with an update frequency of 10 Hz. The policy outputs are continuous velocity actions a_t^r (linear v_t^r and angular w_t^r velocities), reflecting the complex nature of the decision-making process of nonholonomic robots in constrained environments. Fig. 2 shows the network architecture used for the actor and the critic in the MAPPO.

Robot State, s_t^r: The state representation for each robot comprises the robot's angular and heading velocities, roll and pitch. Roll and pitch are considered to facilitate the transfer of learned behaviours to different terrain scenarios in future research.

Robot Perception, I_t^r: For training robots in the PyBullet simulator, each robot perception system considers a binary occupancy map characterised by an 8 m $\times$ 8 m map image centring the robot with a resolution of 0.1m. Within this map, a value of 0 denotes an unoccupied cell including any gap in congestion, while a value of 1 signifies an occupied cell. The positioning and orientation of the occupancy map are synchronised with the x, y translation, and yaw rotation of the robot, with the robot's centre coinciding with the centre of its occupancy map. Since real sensors exhibit some noise, we introduced partial intermittent visibility noise into our occupancy map input during training. For testing in Gazebo and real-world environments, the perception stack (Wildcat sensor-pack [1]) generates a 2D occupancy map from a 3D LiDAR image using OHM mapping [1], with unoccupied or unknown scenario cells converted to 0 and occupied cells to 1.

Give-way Behaviour Policy, π_θ: As function approximators, neural networks with similar architecture were used for a shared policy actor and a centralised critic in the MAPPO framework [18], [20]. Robots acquire cooperative multi-robot interaction behaviours through neural network parameter sharing in a centralised training approach (Fig. 2(A)). The neural networks are not provided with explicit input of other robot's information. Robots are deployed in a decentralised manner using individual robot observations as input to the shared policy (Fig. 2(B)). The utilisation of PPO incorporates clipping to ensure the updated policy does not diverge significantly from the previously learned policy. This approach is critical in preventing catastrophic forgetting, particularly as the policy progresses through various learning curricula. Our training framework used the following reward function:

$$\text{Reward} = R_{\text{goal}} - P_{\text{collision}} - P_{\text{delay}}$$

where, $R_{\text{goal}} = 1000$ was the fixed reward for reaching the goal position. $P_{\text{collision}}$ is the penalty applied when any collision occurs, defined as $0.3 \times P_{\text{delay}}$, where 0.3 and 1000 were determined based on empirical experiments for reward function

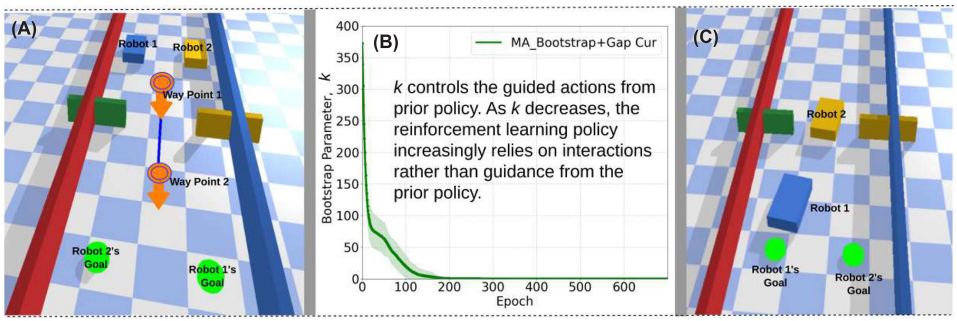

Fig. 3. (A) The reinforcement learning-based give-way MAPPO policy is learning from a prior policy through the bootstrap curriculum. (B) As the give-way behaviour policy gradually learns from the prior policy through the bootstrap curriculum, the influence of the prior policy actions decreases with the reduction of the bootstrap parameter k (see Algorithm 1). This allows the give-way MAPPO to adjust its actions based on guided actions from the prior policy to achieve success. (C) Once the k reaches zero, actions from the prior policy are no longer followed. At this stage, the learning phase transitions to the *gap curriculum*.

tuning. P_{delay} is the delay penalty, accumulating negatively with each timestep that passes, thereby decreasing the total reward by 1 for each unit of time elapsed.

Episode termination is conditioned on either a multi-robot collision or a timeout within a specified local episode length. Also, when both robots are in contact with any obstacle, the episode terminates; otherwise, a collision by only one robot results in a negative reward without ending the episode.

B. Curriculum Learning

Our *multi-robot curriculum* (see 3-B:1) begins by following preprogrammed prior policy actions but progressively transitions to executing its reinforcement learning policy upon meeting the success criteria. The *gap curriculum* (see 3-B:2) incrementally narrows the gaps, thus increasing training difficulty as the success criteria are met. Success in curriculum learning is defined by an agent reaching the goal five consecutive times without multi-robot collisions.

1) Multi-Agent Bootstrap Curriculum (MA_bootstrap): Prior Policy of Behaviour Demonstration: A prior policy of behaviour demonstration was developed through a set of preprogrammed moves to guide the primary give-way behaviour in a multi-robot navigation scenario of lesser complexity (passing a 2 m gap) compared to the ideal constrained scenario (0.85 m gap). Two pre-defined waypoints were established on either side of the congested gap to precisely dictate the robot's movement through this constrained space, as shown in Fig. 3.

In our prior policy, robots were programmed to navigate sequentially toward waypoint 1, followed by waypoint 2, and finally to their designated goal positions, ensuring proper alignment while passing through the gap. We assume robots accurately detect objects up to a range of 2 m. When robots approach side-by-side, one randomly chosen robot will reverse to allow the other robot to lead.

Bootstrapping Prior Policy to Reinforcement Learning Policy: In our training framework, robots receive prior policy actions, $\pi_{prior}(a_t|s_t)$ and PPO policy actions $\pi_\theta(a_t|o_t)$, which are bootstrapped using the following Algorithm 1. Here, we define a bootstrap parameter (k) with an initial value of 400, which is updated based on the achieved curriculum success (C) as shown in Algorithm 1. The k value was tuned and was determined to be 400 based on empirical experiments.

In the initial phase, the *MA_bootstrap curriculum* offers guided actions, aiding agents in learning to navigate through the restricted state space towards the goal with minimal exploration. However, this prior policy is effective only up to a 2 m wide gap, due to the difficulty of tuning multi-robot actions through constrained gaps. Consequently, to accommodate narrower constraints, a supplementary *gap curriculum* is used to decrease the gap to just wider than one robot (0.85 m).

Algorithm 1: Multi-Agent Bootstrap Curriculum.

Given: MAPPO Policy π_θ, Prior Controller π_{prior}, Bootstrap Parameter k, Goal Completion G, Curriculum Success C
Input: MAPPO policy input observation o_t, Prior controller input s_t
Output: MAPPO policy actions $\pi_{bootstrap}(a_t|o_t)$, Prior controller action $\pi_{prior}(a_t|s_t)$
Bootstrapped actions:
$\pi_{bootstrap}(a_t|o_t) = \pi_\theta(a_t|o_t) + \pi_{prior}(a_t|s_t) * k/k_i$
Updating bootstrapped actions using bootstrap curriculum:
Updating bootstrap parameter k based on C
if $C_t = 5$ **then**
  $k \leftarrow 0.75 * k$ { 0.75 was determined based on empirical experiments}
  $C_t \leftarrow 0$
end if
Counting curriculum success C based on Goal completion G
if $G_t = 1$ **then** $C_t \leftarrow C_{t-1} + 1$ **else** $C_t \leftarrow 0$ **end if**

2) Gap Curriculum: The *Gap curriculum* modulates training difficulty to manage congestion effectively when navigating through narrow gaps. We set a decrement rate of 0.1 m each time the curriculum success criteria are met, progressively reducing the gap width from 2 m to 1 m; and a decrement rate of 0.05 m for the gap reduction from 1 m to 0.85 m. This enables substantial exploration and adaptation to the challenging congestion. Fig. 4 shows how give-way behaviour policy learns progressively from a wider gap to a more constrained gap using our *Gap curriculum*.

C. Benchmark Methods

We compare our give-way behaviour policy against four benchmark methods: End-to-end reinforcement learning framework, Hybrid A* [11] and two rule-based algorithms - Backup and Random Wait policy and Right-Hand Road-Rule policy. These rule-based algorithms rely on the Gap policy by Tidd et al. [2] as inherent navigation control towards the goal positions, supplemented by rules to address the multi-robot give-way behaviour, which the single robot Gap policy alone cannot solve. For simplicity, we allow the rule-based benchmark algorithms to know the robot's ID during detection to prevent distinguishing whether it is a static or dynamic object. They capture simple

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

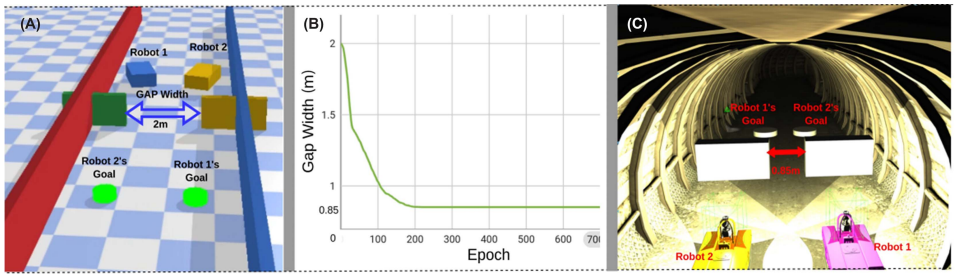

Fig. 4. Training *Gap curriculum* (A) Starting with 2 m gap width (B) Reducing Gap width through curriculum learning success achieved (C) Testing in 0.85 m reduced gap in subterranean tunnel environment developed using the Gazebo simulator.

Algorithm 2: Backup and Random Wait Policy (BRW).

```
while not at goal do
  Update occupancy map I_t^r via sensor readings
  if any robot detected to the side or front in I_t^r then
    Move reverse for 0.5 s then
    Wait for a random duration between 0-6s
  else
    Follow Gap behaviour policy for 0.5s
  end if
end while
```

Algorithm 3: Right-Hand Road-Rule Policy (RR).

```
while not at goal do
  Update occupancy map I_t^r via sensor readings
  if any robot detected to the right side or front in I_t^r
  then
    Move reverse for 0.5s
  else
    Follow Gap behaviour policy for 0.5s
  end if
end while
```

give-way heuristics to determine if our approach is overly complex for the target task. All benchmark methods were tested with the same update frequency (10 Hz) as our original method.

End-to-end reinforcement learning framework: This benchmark uses our MAPPO framework with the same reward function but does not incorporate curriculum learning. Therefore, no initial policy guides the end-to-end reinforcement learning policy, which begins training directly at a 0.85 m wide gap.

Backup and Random Wait Policy (BRW): This benchmark adopts the halting method principle [21] with a local occupancy map, enabling robots to reverse upon detecting a collision and then wait a random period to prevent simultaneous gap approaches. The detailed implementation of the BRW policy is provided in Algorithm 2.

Right-Hand Road-Rule Policy (RR): Inspired by Australian traffic standards, this benchmark adopts a right-hand road-rule approach to give way to any robot detected on the right side, similar to the rule-based benchmark used by Park et al. [21]. Algorithm 3 shows the RR policy rule.

Hybrid A:* The benchmark, Hybrid A* can recalculate edge cost dynamically based on a 3D sensor map, which enabled multi-robot navigation in the DARPA Subterranean Challenge by team CSIRO [11]. We have used the Hybrid A* algorithm as our baseline and tested integrating with ROS Navigation Stack in the Gazebo simulator in a subterranean tunnel environment.

IV. Experiments

In our experimental setup for training and testing simultaneous multi-robot navigation through constrained pathways, two robots were positioned side-by-side, 4 m away from the gap and 8 m from their goal positions. Goal positions were randomly assigned on the opposite side of the gap, ensuring that the robots' paths crossed to create conflict scenarios. The robots' initial positions were also randomised with an inter-robot distance between 1-3 metres to encourage generality. The task environment was established within a 4.5-metre-wide tunnel to constrain exploration. Training results are discussed in Section V-A.

We supported our contributions by testing our give-way behaviour policy respectively in fast, and high-fidelity simulators: PyBullet and Gazebo, to validate performance and transferability in SIM-to-SIM scenarios. The Gazebo Simulator test environment replicated an underground subterranean tunnel, developed by Team CSIRO for the DARPA Subterranean Challenge, as depicted in Fig. 4(C). Finally, we conducted an open-field test on a curvilinear bitumen road. The field test area featured two bollards as static obstacles, placed to create different constrained gaps (1.5 m, 1 m, 0.85 m) within a 3-metre-wide road flanked by grass, which were recognised as barriers in the 2D occupancy maps produced by LiDAR sensors (see Section V-B).

An important aspect of these experiments is the emerging behaviours for multi-robot give-way interactions. These interaction behaviours were compared with benchmarks in the simulation and field tests, enabling discussion of their benefits and real-world applicability (see Section V-C).

V. Results

A. Training Results and Comparison With End-to-End Learning

The training of the give-way behaviour policy using our proposed *MA_bootstrap curriculum* and *Gap curriculum* based reinforcement learning was conducted in the PyBullet simulator. This approach was compared with the end-to-end reinforcement learning policy trained without any curriculum. The training experiments were conducted across 30 trials to ensure statistical significance in the results. Each trial was executed on 64 CPUs in parallel, where each CPU processed 700 epochs of five episodes each and updated the gradients through averaging. Fig. 5 demonstrates that our proposed curriculum learning techniques facilitate progressive learning, whereas no performance gain was achieved using only the end-to-end reinforcement learning or only the *MA_bootstrap curriculum* (displayed similar plot to end-to-end learning plot) using our reward function. The results showed suboptimal performance when training the reinforcement learning policy with the *Gap curriculum* alone.

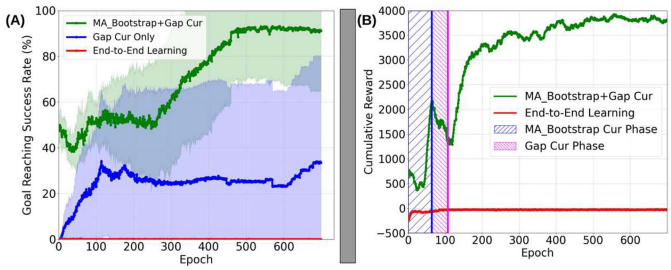

Fig. 5. (A) Training performance comparison among the give-way MAPPO, end-to-end learning and a policy using *gap curriculum* only. Goal-reaching success was measured as the percentage of successful navigation, assessed through parallel processing on a 64-CPU system. (B) Highlighting performance improvement through curriculum learning phases in the cumulative reward curve of a training trial.

TABLE I
BENCHMARK COMPARISON IN SIMULATIONS: SUCCESS IS DEFINED AS THE PERCENTAGE OF TRIALS WHERE BOTH ROBOTS REACHED THE GOAL WITHOUT COLLIDING

Method	Gap (m)	R C	O C	Success (%)	Mean TTG (R_L&R_F)	Std TTG (R_L&R_F)
Give-way MAPPO	1.50	0	0	**100**	**9.5 & 11.9**	**0.063 & 0.063**
	1.00	0	0	**100**		
	0.85	0	1	**99**		
BRW	1.50	0	0	**100**	9.6 & 16.6	0.068 & 0.069
	1.00	0	5	95		
	0.85	0	15	85		
RR	1.50	0	0	**100**	9.5 & 16.5	0.067 & 0.065
	1.00	0	4	96		
	0.85	0	16	84		
Hybrid A*	1.50	30	0	0	N/A	N/A

Time to goal is reported separately for the leading and following robots. Benchmarks BRW and RR were tested in the PyBullet simulator for 100 trials each. The Hybrid A* algorithm was tested in the Gazebo simulator due to its reliance on 3D sensor maps and was not evaluated in real-world scenarios to avoid safety breaches related to multi-robot collisions. The number of test trials for Hybrid A* was limited to 30 due to the higher computational demands in Gazebo. RC and OC indicate inter-robot collision and collision with any obstacle, respectively. Time to Goal (TTG) is measured as mean and standard deviation for the leading robot (R_L) and the following robot (R_F).

Training started considering a 2 m gap with an initial reward due to guided actions from the *MA_bootstrap curriculum*. As training progressed, the bootstrap parameter was gradually reduced after meeting the success criterion (Fig. 3(B)). Once agents had fully learned the prior policy behaviour, the training phase transitioned to the *Gap curriculum*, leading to an initial performance drop due to the limited optimal trajectories available to goal positions through small gaps. Subsequent training using the *Gap curriculum* involved agents navigating through progressively narrower gaps, down to 0.85 m. This leveraged their prior experience to enhance performance in avoiding obstacles and reaching goals. In contrast, end-to-end reinforcement learning with our simple reward function failed to address the high exploration challenge and, therefore, did not learn an effective policy for our targeted scenario, due to slow convergence.

B. Test Results With Benchmark Comparison

Based on training results, the best-performing give-way behaviour policy, guided by *MA_bootstrap* and *gap curricula* over 30 trials, was tested and benchmarked in simulations which are shown in Table I.

To evaluate our model's ability to operate with actual sensor data and to test its robustness in bridging the simulation-to-reality gap, as well as its generalisability in handling real-world uncertainties, we conducted tests in the Gazebo simulator using sensor data and also ran field robot tests at various wide gaps, as shown in Table II.

TABLE II
GIVE-WAY MAPPO RESULTS IN GAZEBO (SIMULATED SUBTERRANEAN TUNNEL ENVIRONMENT) AND FIELD TESTS

Test Env.	Gap (m)	# Trials	R C	O C	Success (%)	Mean TTG (R_L&R_F)	Std TTG (R_L&R_F)
Gazebo	1.50	30	0	0	100	9.5 & 13	0.120 & 0.363
	1.00		0	2	93.3		
	0.85		0	8	73.3		
Field Test	1.50	10	0	0	100	10 & 14	0.126 & 0.352
	1.00		0	3	70.0		
	0.85		0	4	60.0		

C. Behaviour Findings in Simulated & Real-World

During the testing of our model, several emergent robot behaviours were observed in multi-robot give-way scenarios when navigating constrained gaps. We analyse the behaviours learned by the neural network by replicating potential occurrence scenarios. We often preprogrammed one robot to create a conflicting scenario that induced emerging behaviour from the other robot using our give-way MAPPO policy. Fig. 6 illustrates the behaviour trajectories by replicating the corresponding scenarios tested in the PyBullet simulator.

In Fig. 6(A), two robots started side-by-side from a close 1.2 m distance using give-way MAPPO policy. One robot exhibited a reverse movement between 1 s and 2 s (marked by a red circle), which can be termed a *close call* behaviour to avoid a potential multi-robot collision. Additionally, the *close call* performing robot was observed following the leading robot near the gap conflict zone showing the *following* behaviour (marked by a pink hatch). In Fig. 6(B), both robots initiated our give-way MAPPO and displayed *give-way* behaviour. After one robot positioned itself behind the other to follow, the leader was preprogrammed to reverse for 3-8 seconds (marked by a red circle on the blue trajectory). During this time, the following robot showed *cooperative reverse driving* behaviour (marked by a red circle in the yellow trajectory). In Fig. 6(C), both robots were initially programmed to move straight toward the gap for the first 2s. Then our give-way MAPPO policy activated and both robots made a sharp turn, achieving *gap alignment* behaviour to pass through the gap. Notably, the following robot slightly reversed to orient itself properly, exemplifying the *gap alignment* behaviour. Fig. 6(D) is an extended give-way scenario. Both robots initiated our give-way MAPPO and displayed *give-way* behaviour. After one robot positioned itself behind the other to follow, the leading robot was preprogrammed to be stopped for 5 s between 3–8 s periods. During this time, the following robot exhibited an extended *give-way* or *stopping* behaviour using our policy, stopping for 5 s, a behaviour particularly advantageous for robot safety in case of malfunction or stoppage of the leading robot. The behaviours explained in Fig. 6 were also observed in the field tests shown in Fig. 7.

VI. DISCUSSIONS AND LIMITATIONS

Our give-way MAPPO system is guided by the *MA_bootstrap curriculum* and further refines training scenario complexity using the *Gap curriculum*. Our methodology helps in finding reward signals and addresses slow convergence, which was responsible for the end-to-end reinforcement learning method's

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

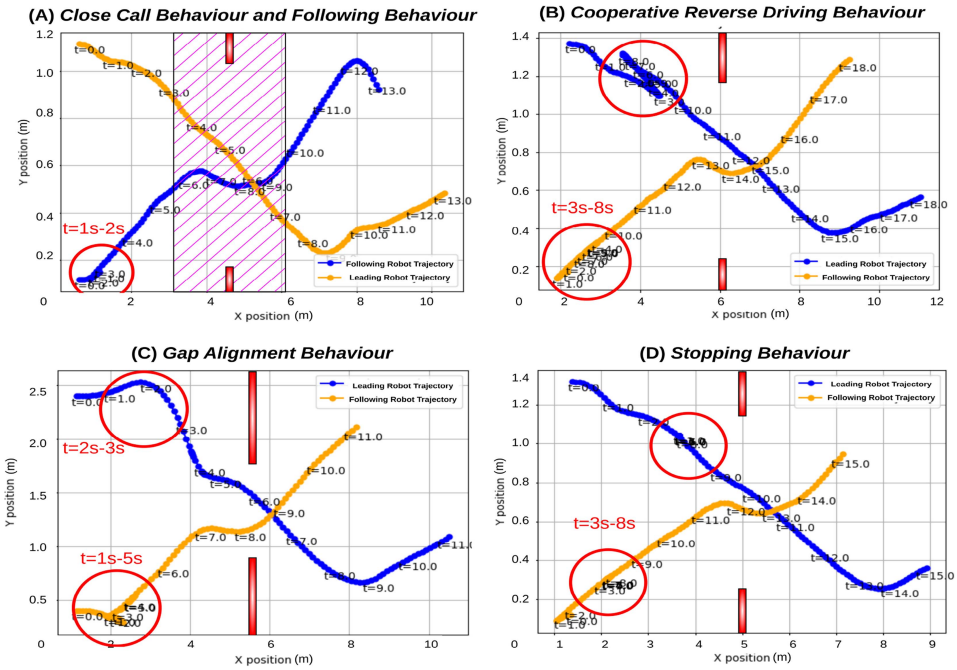

Fig. 6. Gallery of robot trajectories highlighting emergent behaviours during simultaneous cross-navigation through a 0.85 m gap.

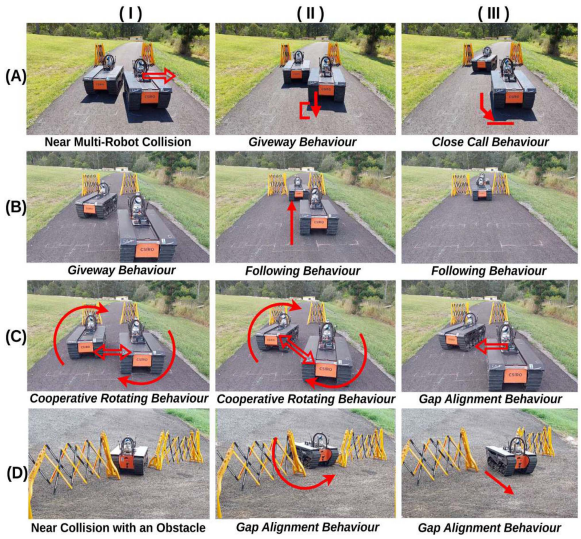

Fig. 7. Gallery of Behaviours: (A) Two robots simultaneously approached a potential conflict scenario: one robot executed a right turn to initiate *give-way* behaviour and subsequently reversed to avoid a collision. Such movements demonstrated a *close call* behaviour observed frequently during field tests, especially when robots commence from proximate starting positions. (B) This scenario captured one robot adopting the *following* behaviour, trailing the lead robot through the gap after a period of yielding. (C) The robots exhibited *cooperative rotating* behaviour, with one robot aligning its heading toward the gap and taking the lead while the other robot maintained a cautious relative distance to prevent possible close calls and followed the leading robot. (D) After demonstrating the *give-way* behaviour, the following robot adjusted its heading using the *gap alignment* behaviour. This behaviour involved the robot approaching closely to an obstacle corner and then rotating to an optimal orientation for gap navigation. Moreover, *cooperative reverse* behaviour was observed in the field test. After showing give-way, the leading robot reversed to employ gap alignment behaviour, and the following robot also reversed, ensuring a relatively safe distance greater than that observed in *closecall* scenarios.

failure to meet high search demands in restricted spaces with a simple reward function. Employing a complex, sparse reward could potentially enhance performance in end-to-end reinforcement learning. However, excessive tuning might compromise generalisation and prove time-consuming to learn to navigate through constrained scenarios. In contrast, the integration of both *MA_bootstrap* and *Gap curricula* significantly enhances training outcomes as guided exploration leads to appropriate reward signalling and progressive learning. In comparison, sole reliance on the *Gap curriculum* is associated with a greater likelihood of suboptimal results (Fig. 5(A)). Any controller that guides the agents to a reward signal through a basic collision-free navigation strategy can be used as the bootstrap prior policy.

Our model shows consistent give-way behaviour and emerging interaction behaviours in simulated (Fig. 6) and real-world environments (Fig. 7). Such emerging behaviours are crucial for ensuring precise, collision-free multi-robot interaction in narrow gaps, including in unprecedented scenarios such as robot malfunctions (Fig. 6).

Despite our method using a shared policy to help mitigate nonstationarity and partial observability [16], real-world scenarios may often suffer from partial observability. Since decentralised execution does not consider other robots' states, agent policies are assumed to be independent during decentralised execution [22]. Employing DEC-POMDP could enhance field test performance by improving partial observability. However, creating a globally optimal model-free MARL method for DEC-POMDP remains an open research problem [23].

Our experimental results (Table I) show that our policy achieved a 99% success rate in the PyBullet simulator. However, performance decreases in Gazebo and real-world tests (Table II) due to sensor noise, surface friction, and other real-world uncertainties, which also contribute to a high standard deviation in time to goal. The BRW and RR benchmarks exhibit delays in

reaching the goal positions for the following robot, ranging from 16.5–16.6 s, compared to 11.9 s under our give-way MAPPO. Rule-based approaches show higher collision rates by driving the controller into out-of-distribution states, which could be improved through more generalised training of the inherent controller. They also lack effective multi-robot interaction behaviours in the scenarios depicted in Figs. 6 and 7. Furthermore, the BRW and RR benchmarks show undesirable oscillatory behaviours when tested in *closecall* and extended *stopping* behaviour scenarios, which increased delays in reaching the goal positions. While the Hybrid A* algorithm fails to recalculate dynamic edge cost for a simultaneous approach towards a 1.5 m wide gap or below, no collisions between robots were observed in any test scenario when using our give-way MAPPO policy.

Future research is needed to identify the conditions that should activate and deactivate the multi-robot give-way behaviour. Increasing the number of robots in our scenario will increase the computational costs of training. Our framework is tested for homogeneous robots. Reducing sensor noise and generalising our policy to relevant scenarios will enhance real-world performance.

VII. Conclusion

This research presents a novel curriculum-based multi-robot reinforcement learning framework that can develop collision-free navigation policies in highly constrained gap environments. Our framework incorporates the *MA_bootstrap curriculum* and *gap curriculum* for developing a give-way behaviour policy that also learns emerging multi-robot interaction behaviours. These behaviours are difficult to manually replicate or deduce from existing knowledge for effective constrained navigation. The rule-based benchmark methods lack interaction behaviours, so there are disadvantages due to delayed time to reach goal positions. Our give-way MAPPO shows superior performance with higher success rates: nearly 99% in PyBullet, 73% in Gazebo, and 60% in the real-world through a 9.0% wide gap (0.85 m) considering robot width (0.78 m). The failure resulted not from inter-agent collisions but from collisions while navigating through the gap. This issue could be addressed by improving sensor noise and policy generalisation in real-world domains. Our model is suitable for decentralised multi-robot navigation using only local sensors in GPS-denied search and rescue operations. It can also coordinate homogeneous robots from different companies with different control frameworks without relying on inter-agent communication.

Acknowledgment

The authors would like to thanks to Brett Wood, Fletcher Talbot, Tom Hines, and Ross Fiamingo for helping run field operation tests and ensuring the safety of the experimental environment.

References

[1] N. Kottege et al., "Heterogeneous robot teams with unified perception and autonomy: How team CSIRO Data61 tied for the top score at the DARPA subterranean challenge," in *IEEE Trans. Field Robot.*, vol. 2, pp. 100–130, 2023, DOI:10.1109/TFR.2024.3522063.

[2] B. Tidd, A. Cosgun, J. Leitner, and N. Hudson, "Passing through narrow gaps with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*. Prague, Czech Republic, 2021, pp. 3492–3498. [Online]. Available: https://ieeexplore.ieee.org/document/9812263/

[3] Y. Yan et al., "Relative distributed formation and obstacle avoidance with multi-agent reinforcement learning," in *Proc. Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, 2022, pp. 1661–1667. [Online]. Available: https://ieeexplore.ieee.org/document/9812263/

[4] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of A* pathfinding," *Procedia Comput. Sci.*, vol. 179, pp. 507–514, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050921000399

[5] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *J. Guidance, Control, Dyn.*, vol. 25, no. 4, pp. 755–764, Jul. 2002. [Online]. Available: https://arc.aiaa.org/doi/10.2514/2.4943

[6] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996. [Online]. Available: https://ieeexplore.ieee.org/document/508439/

[7] S. M. LaValle, "Rapidly-exploring random trees : A new tool for path planning," *Annu. Res. Rep.*, Computer Science Department, Iowa State University, tech. rep. TR 98-11, 1998.

[8] M. Hamer, L. Widmer, and R. D'andrea, "Fast generation of collision-free trajectories for robot swarms using GPU acceleration," *IEEE Access*, vol. 7, pp. 6679–6690, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8587164/

[9] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, Jun. 2020. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364920916531

[10] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *J. Mach. Learn. Res.*, vol. 21, no. 181, pp. 1–50, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-212.html

[11] T. Hines et al., "Virtual surfaces and attitude aware planning and behaviours for negative obstacle navigation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 4048–4055, Apr. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9376244

[12] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007. [Online]. Available: https://ieeexplore.ieee.org/document/4118472/

[13] D. Chen, H. Li, Z. Jin, H. Tu, and M. Zhu, "Risk-anticipatory autonomous driving strategies considering vehicles' weights, based on hierarchical deep reinforcement learning," in *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 12, pp. 19605–19618, Dec. 2024, DOI:10.1109/TITS.2024.3458439.

[14] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat. Proc.*, 1991, pp. 1398–1404. [Online]. Available: https://ieeexplore.ieee.org/document/131810

[15] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist. JMLR Workshop Conf. Proc.*, 2011, pp. 627–635. [Online]. Available: https://www.marl-book.com/#cite

[16] S. V. Albrecht, F. Christianos, and L. Schäfer, "Multi-agent reinforcement learning: Foundations and modern approaches," 2024. [Online]. Available: https://www.marl-book.com/#cite

[17] S. Yao, G. Chen, L. Pan, J. Ma, J. Ji, and X. Chen, "Multi-robot collision avoidance with map-based deep reinforcement learning," in *Proc. IEEE 32nd Int. Conf. Tools Artif. Intell.*. Baltimore, MD, USA, 2020, pp. 532–539. [Online]. Available: https://ieeexplore.ieee.org/document/9288300/

[18] A. Smit, H. A. Engelbrecht, W. Brink, and A. Pretorius, "Scaling multi-agent reinforcement learning to full 11 versus 11 simulated robotic football," *Auton. Agents Multi-Agent Syst.*, vol. 37, no. 1, Mar. 2023, Art. no. 20. [Online]. Available: https://doi.org/10.1007/s10458-023-09603-y

[19] J.-S. Park, X. Xiao, G. Warnell, H. Yedidsion, and P. Stone, "Learning perceptual hallucination for multi-robot navigation in narrow hallways," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10033–10039. [Online]. Available: https://ieeexplore.ieee.org/document/10161327

[20] S. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel, "Emergent coordination through competition," in *Proc. Int. Conf. Learn. Representations*, New Orleans, LA, USA, May 2019. [Online]. Available: https://openreview.net/forum?id=BkG8sjR5Km

[21] J. S. Park, B. Tsang, H. Yedidsion, G. Warnell, D. Kyoung, and P. Stone, "Learning to improve multi-robot hallway navigation," in *Proc. Conf. Robot Learn.*, 2021, pp. 1883–1895. [Online]. Available: https://proceedings.mlr.press/v155/park21a.html

[22] Y. Zhou et al., "Is centralized training with decentralized execution framework centralized enough for MARL?," May 2023, *arXiv:2305.17352*.

[23] C. Amato, "An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning," Sep. 2024, *arXiv:2409.03052*.