Deep Multi-Agent Reinforcement Learning for Multi-Robot Social Navigation in Constrained Environments

Takieddine Soualhi[1], Jacques Saraydaryan[2], Laetitia Matignon[3]

***Abstract*— Developing effective robot navigation methods in constrained environments with pedestrians is essential for real-world applications. Although recent deep reinforcement learning methods have shown promising results in addressing the problem of social navigation, they often focus solely on single-robot scenarios and overlook the presence of static obstacles in the environment. In this work, we address the problem of multi-robot social navigation in constrained environments. We introduce an environment specifically designed for multi-robot social navigation, allowing the simultaneous simulation of a fleet of robots alongside humans and static obstacles. In addition, we propose a novel method for learning socially aware multi-robot navigation policies in constrained environments using multi-agent reinforcement learning. Our approach leverages convolutional and graph neural networks to learn structured representations of scene geometry as well as the interactions between robots and humans in the environment. Experimental results demonstrate that our approach outperforms existing single-robot social navigation baselines and enables efficient, implicit coordination in heterogeneous crowds while effectively accounting for static obstacles in the environment.**

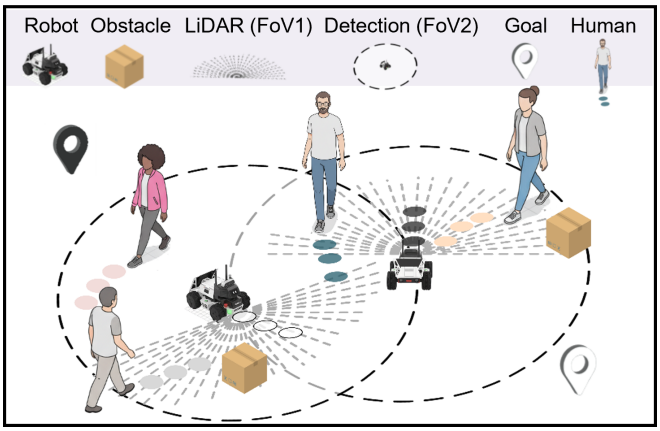

Fig. 1: Illustration of our navigation task and the proposed approach: each robot aims to reach its goal using a LiDAR sensor with field of view (FoV)1 to perceive scene geometry, and a detection system with FoV2 to detect other entities

I. Introduction

Robot navigation in crowded environments has emerged as a distinct research field in response to the growing deployment of service robots and autonomous vehicles in human environments. Beyond simply reaching their destinations, autonomous robots must integrate socially appropriate behaviors such as maintaining personal space, adapting speed to pedestrian flow, and signaling intent through smooth trajectories to perform tasks efficiently and seamlessly around humans [1].

Over the past decade, learning-based approaches have emerged as a category of methods that reduce reliance on manual modeling by training robot control policies through deep reinforcement learning (DRL). These approaches typically make minimal assumptions about human motion and instead train control policies in simulation, thereby shifting the complexity of the task to the neural network [2]. Prior work on DRL-based methods has explored diverse techniques for robot crowd navigation. Yet existing approaches suffer from one or a combination of three key limitations: (i) they assume agents operate in obstacle-free, open environments [2], [3], (ii) they employ scene representations and navigation algorithms that do not distinguish between dynamic and static obstacles [4], (iii) they assume only scenarios with a single robot in the scene [5], [6]. Consequently, these assumptions limit the feasibility of such methods in real-world scenarios.

In this work, we address all of these challenges by using multi-agent reinforcement learning (MARL) [7] to tackle the problem of multi-robot social navigation in constrained environments (Fig. 1). First, we propose a training environment that extends the CrowdNAV simulator [8] to multi-agent settings, enabling the training of multiple robots for social navigation tasks. We further adapt the simulator by adding 2D LiDAR sensing and incorporating static obstacles, thereby supporting the simulation of constrained environments. We then introduce HAMRON (Human Aware Multi-Robot Navigation), a DRL method for learning multi-robot navigation strategies in crowded, constrained environments. The main challenges, compared to the existing methods, include learning coordinated, human-safe navigation strategies for a fleet of robots and managing interactions with both controlled entities (robots) and uncontrolled entities (humans and static obstacles).

Inspired by recent work on single-robot LiDAR-based navigation [5] and GNN-based social navigation [9], HAMRON employs a GNN encoder with attention to model heterogeneous interactions between a fleet of robots and the surrounding human crowd, and a CNN-based encoder to capture the spatial geometry of the scene. The resulting

[1] Inria, CITI Lab, INRIA–INSA Chroma Team, Villeurbanne, France. takieddine.soualhi@inria.fr

[2] CPE Lyon, CITI Lab, INRIA–INSA Chroma Team, Villeurbanne, France. jacques.saraydaryan@cpe.fr

[3] Univ Lyon 1, INSA Lyon, CNRS, LIRIS, UMR 5205, Villeurbanne, France. laetitia.matignon@univ-lyon1.fr

This work was funded by the French National Research Agency under the France 2030 program, reference ANR-23-DMRO-0018.

representations from both modules are used to train individual robot control policies within a MARL framework under realistic assumptions about perception and communication. During inference, each robot relies solely on its local sensors, and no information is shared between robots.

This paper is organized as follows: Section II presents the task setup and proposed approach. Section III describes the experimental setup. Section IV reports and discusses the results. Section V concludes the paper.

II. Multi-robot Social Navigation in Constrained Environments

In this section, a formulation of the multi-robot social navigation problem within MARL is presented, and our proposed approach is introduced.

A. Problem statement

Consider a fleet of N_r robots initialized in a scene along with a crowd of N_h humans and N_o static obstacles. In addition to avoiding static obstacles, each robot i has to reach its goal position g^i without relying on any prior map and by maintaining socially compliant behavior with respect to other entities (robots and humans). It is important to note that we assume robots do not share any information for navigation. Each robot relies solely on its local perception system, composed of 2D LiDAR scans l_t^i with FoV1 and a perception module that captures the positions p_t^j of other entities within FoV2 with $j \in [1, ..., (N_r+N_h)]$ and t denotes the timestep. Using different FoVs for the robot's perception allows for the usual constraints of real robots to be taken into account. The Markov decision process elements (i.e., the action space, observation space, and reward function) for this task are defined in the following sections.

a) Action space: We adopt a holonomic kinematic model where, at each instant t, robot i executes $a_t^i = (v_{x,t}^i, v_{y,t}^i)$ with planar linear velocities $v_{x,t}^i, v_{y,t}^i$ in the local frame.

b) Observation space: At time t, robot i receives an observation $o_t^i = \{l_t^i, w_t^i, G_t^i\}$ with three components: (i) 2D LiDAR scans l_t^i for perceiving static obstacles, (ii) an intrinsic state vector $w_t^i = [p_t^i, v_t^i, g^i, \phi_t^i, \rho_t^i]$ encoding the robot position, velocity, goal, yaw, and radius, and (iii) an interaction graph G_t^i representing perceived humans and robots. For each robot i, we define $G_t^i = (\mathbf{M}_t^i, \mathbf{E}_t)$, where $\mathbf{M}_t^i$ is the visibility matrix over entities detected within its field of view, and $\mathbf{E}_t$ contains node features e_t^j for each entity (Fig. 2). Each node feature includes a type label (e.g., $label \in \{\text{robot}, \text{human}\}$) and a short predicted trajectory $T_t^j = [p_t^j, \dots, p_{t+5}^j]$ expressed in the robot local frame. Future positions are estimated using a constant-velocity model from consecutive observations, which we found sufficient for short-horizon prediction.

c) Reward function: The reward is designed to promote safe and efficient multi-robot navigation. For robot i, it is defined as

$$r_t^i = \begin{cases} r_c & \text{if } i \text{ collides,} \\ r_s & \text{if } i \text{ reaches its goal,} \\ r_{pot,t}^i + r_{pred,t}^i & \text{otherwise,} \end{cases} \tag{1}$$

where r_c penalizes collisions with humans, robots, or static obstacles, and r_s rewards goal reaching. To discourage entering the predicted paths of other entities, we define

$$\begin{aligned} r_{pred,t}^{i,j} &= \min_{k\in[1,5]} \left(\mathbb{1}_{i,j}^{t+k} \frac{r_c}{2^k} \right), \\ r_{pred,t}^{i} &= \min_{j\in[1,\dots,(N_r+N_h)]} r_{pred,t}^{i,j}, \end{aligned} \tag{2}$$

where $\mathbb{1}_{i,j}^{t+k}$ indicates whether agent i collides with the k-th predicted position of entity j. This term penalizes near-future intrusions more strongly than distant ones. Finally, progress toward the goal is encouraged with the potential-based reward

$$r_{pot,t}^i = -d_{g,t}^i + d_{g,t-1}^i, \tag{3}$$

where $d_{g,t}^i$ is the distance from robot i to its goal at time t.

B. Approach

In this section we present the DRL model of HAMRON, illustrated in Fig. 2. First, LiDAR scans l_t^i are encoded by a 4-layer 1D CNN encoder with kernels of varying sizes, enabling the model to learn representations of obstacles across multiple scales. This 1D CNN extracts spatial features that capture scene geometry, a design choice motivated by established 2D LiDAR-based navigation methods [10] and the SOTA single-robot social navigation approach [5], which show that combining 1D CNNs with recurrent policies yields strong obstacle avoidance performance. Conversely, the intrinsic state information w_t^i of robot i is encoded using a multi-layer perceptron (MLP) into a feature vector that captures its state. In parallel, the interaction graph G_t^i is passed through an interaction filter that applies a GAT layer followed by a Gumbel–Softmax mask to the nodes corresponding to entities within FoV2 of robot i, producing a sparse graph $G_{s,t}^i$. This allows the policy to endogenously learn to attend to the most critical agents in dynamic and unpredictable crowds, effectively filtering out noise and focusing on the entities that pose the greatest risk to navigation safety and efficiency. A GAT layer is then applied to this sparse graph to compute node features influenced by the filtered neighboring entities, yielding the crowd graph $G_{c,t}^i$. Intuitively, the robot needs to consider only relevant crowd entities, since not all interactions are equally important. From the crowd graph, the node corresponding to robot i is extracted and concatenated (CAT) with the robot's spatial and intrinsic features to form a joint representation for downstream policy learning. Finally, a gated recurrent unit (GRU) followed by two MLPs outputs a value estimation and control action, conditioned on the previous hidden state h_{t-1}^i of the GRU and the joint representation.

For MARL training, we adopt a decentralized training and execution paradigm, where each robot learns from its own

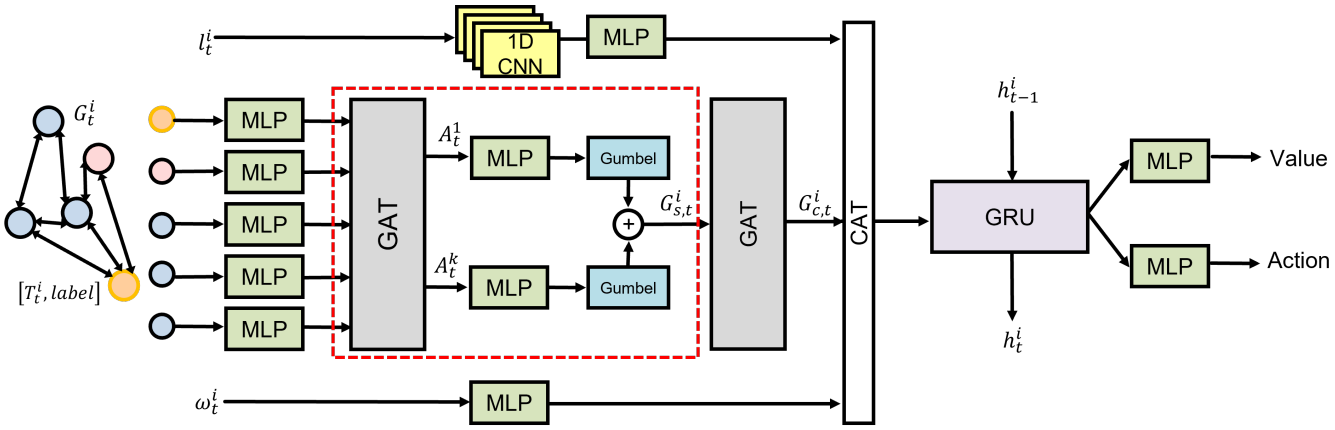

Fig. 2: Illustration of the proposed DRL model: for a given agent i (orange node), the input includes intrinsic information, LiDAR scans within FoV1, and a graph restricted to FoV2. Each node (pink for other robots and blue for other humans) contains the observed entity's current position, predicted positions, and a label identifying its class. The red dashed line in this figure highlights the interaction filter.

local observations and reward. We use parameter sharing across robots and deploy the learned policy independently on each robot at test time. Training is performed with IPPO [7], which enables scalable multi-agent learning without a centralized critic.

III. Experimental Setup

A. Simulation Environment

a) Simulator: Our simulator extends the single-agent CrowdNav simulator to the multi-agent setting. Our implementation includes simulation of parameterizable static obstacles (e.g., number and size) modeled as circles. We integrate a LiDAR sensor simulation via ray tracing using the RangeLib library [11], which provides ray-tracing algorithms for 2D LiDAR simulation. The LiDAR sensor can be parameterized by its FoV, range, and number of rays. We also integrate occupancy maps into the simulator, enabling costmap modalities and supporting studies of 2D SLAM approaches.

b) Scenario: We consider scenarios with randomly placed circular static obstacles, humans moving across the scene toward opposite-side goals, and robots initialized with independent random goals. Humans, simulated using a social force (SF) model, react only to other humans and static obstacles, creating an adversarial crowd. An episode ends when all robots have either collided or reached their goals, and collided robots are reset with new goals. Unless stated otherwise, we use $r_c = -1$, $r_s = 10$, FoV2 $= 360°$, FoV1 $= 230°$, and fix the number of static obstacles to $N_o = 15$.

B. Metrics

We evaluate our approach using standard social navigation metrics: success rate, collision rate, intrusion ratio, travel length, and travel time. Success rate measures the proportion of robots reaching their goals, while collision rate measures the proportion colliding with a human, robot, or obstacle. Intrusion ratio is the fraction of time a robot remains within $0.25\,\text{m}$ of a human, following [3]. Travel length and travel time denote the mean path length and mean time to goal across test episodes.

C. Methods

Since we study constrained environments, we compare against DRL baselines that explicitly handle static obstacles. Because these methods are designed for single-robot settings, each policy is trained with one robot and then deployed independently on all robots at test time.

a) HEIGHT [5]: HEIGHT is a strong single-robot baseline for social navigation in constrained environments.

b) LiDAR-Nav [12]: LiDAR-Nav is a recurrent navigation policy based only on 2D LiDAR observations.

c) HAMRON: Although HAMRON is designed for multi-robot training, we also evaluate it in the same single-robot training setup for fair comparison with the baselines. Each method is trained for 10M timesteps with an episode length of 100 timesteps. We note that training is performed in three independent runs with different random seeds. For evaluation, we report the mean of the previously introduced metrics of the three trained models across 100 test episodes, using the same evaluation seed.

IV. Experimental Results

Table I reports comparison results across two evaluation settings: single-robot and multi-robot. In the single robot setting, HAMRON attains the highest success rate while yielding the shortest travel time and length, largely by suppressing timeouts relative to LiDAR-Nav and HEIGHT. However, this comes with a slightly higher collision rate than HEIGHT and an intrusion ratio closer to LiDAR-Nav, indicating a more assertive navigation style. In the multi-robot setting where the number of robots varies between training and testing, LiDAR-Nav remains less effective. It achieves the lowest intrusion ratio but also has the lowest success rate and the highest timeout, suggesting that while

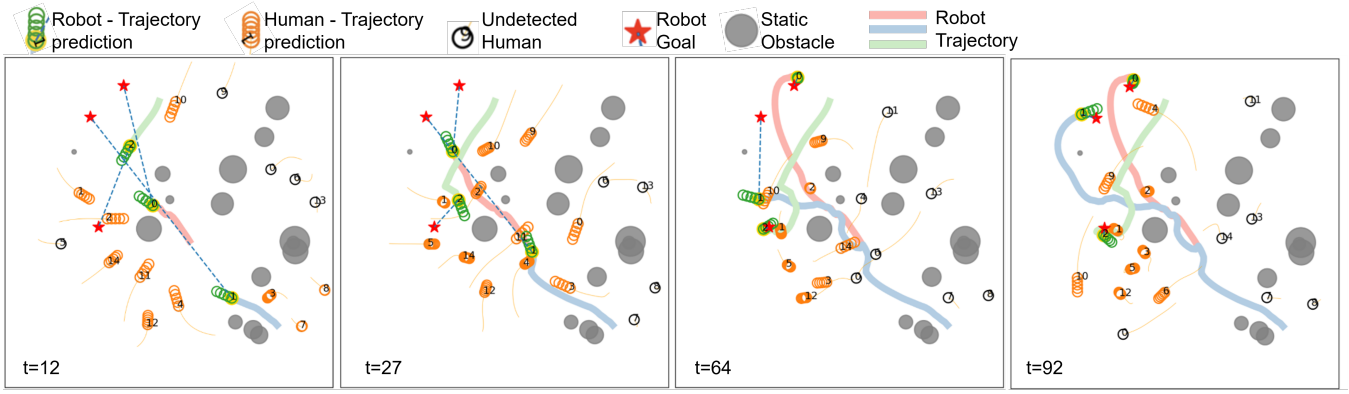

Fig. 3: Illustration of the trajectories performed by three robots under HAMRON across four sequential timesteps, displayed from left to right.

the robots keep their distance from humans, they often fail to reach their goals. We attribute this to the partial observability inherent in LiDAR-based perception, where the robot has to infer the environment state solely from telemetric information. Although HAMRON trained with a single robot still outperforms LiDAR-Nav and HEIGHT on all other metrics when deployed in multi-robot scenarios, methods tailored for single-robot settings, including this variant of HAMRON, experience a drop in performance when deployed in multi-robot environments. This is due to the fact that these models weren't exposed to interactions with other robots during training, leading to poor coordination at deployment. Building on this, MARL training further improves the performance of HAMRON in multi-robot settings (Fig. 3). We suggest that, during training, multi-agent learning allowed the model to learn and capture strategies that account for the behavior of other robots, mitigating coordination failures and deadlocks inherent in single-robot policies. By training in a multi-agent setting, the robots learn implicit social protocol and trajectory anticipation that are essential for navigating heterogeneous crowds where human and robot intentions are often unpredictable.

Method	Success ↑	Collision ↓	Timeout ↓	Intrusion ↓ Ratio	Travel ↓ Time	Travel ↓ Length	Train N_r	Train N_h	Test N_r	Test N_h
LiDAR-Nav [4]	0.60	0.22	0.17	19.64	13.77	14.50				
HEIGHT [3]	0.64	0.20	0.16	15.22	13.51	13.88	1	15	1	15
HAMRON (ours)	0.67	0.28	0.05	19.78	12.26	11.80				
LiDAR-Nav [4]	0.51	0.28	0.21	13.06	14.13	15.00	1	15		
HEIGHT [3]	0.56	0.33	0.11	13.41	12.72	12.40	1	15	3	15
HAMRON (ours)	0.62	0.33	0.05	15.51	12.25	11.93	1	15		
HAMRON (ours)	0.69	0.25	0.05	15.75	12.25	12.03	3	15		

TABLE I: Comparison of our approach with baselines in single and multi-robot scenarios.

V. Conclusion

We introduced HAMRON, a multi-agent reinforcement learning approach for human-aware multi-robot navigation in constrained environments, together with a simulator including multiple robots, humans, and static obstacles. By combining LiDAR-based perception with graph-based interaction modeling under decentralized training and execution, HAMRON yields strong performance compared to single-robot baselines and achieves efficient implicit coordination in multi-robot settings. Our initial results suggest that HAMRON provides a promising framework for navigation in complex, unpredictable scenarios. Future work will include more extensive evaluation, comparisons against dedicated multi-robot baselines, and real-world validation.

References

[1] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core Challenges of Social Robot Navigation: A Survey," ACM T-HRI, 2023.

[2] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in IEEE ICRA, 2017.

[3] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in IEEE ICRA, 2023.

[4] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," SAGE IJRR, 2020.

[5] S. Liu, H. Xia, F. C. Pouria, K. Hong, N. Chakraborty, Z. Hu, J. Biswas, and K. Driggs-Campbell, "Height: Heterogeneous interaction graph transformer for robot navigation in crowded and constrained environments," arXiv:2411.12150, 2024.

[6] Z. Xie and P. Dames, "Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles," IEEE T-RO, 2023.

[7] C. S. D. Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" arXiv:2011.09533, 2020.

[8] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in IEEE ICRA, 2019.

[9] E. Escudie, L. Matignon, and J. Saraydaryan, "Attention graph for multi-robot social navigation with deep reinforcement learning," in AAMAS, Extended Abstract, 2024.

[10] F. Leiva and J. Ruiz-del Solar, "Robust rl-based map-less local planning: Using 2d point clouds as observations," IEEE RA-L, 2020.

[11] C. Walsh and S. Karaman, "Cddt: Fast approximate 2d ray casting for accelerated localization," arXiv:21705.01167, 2017.

[12] H. Beomsoo, A. A. Ravankar, and T. Emaru, “Mobile robot navigation based on deep reinforcement learning with 2d-lidar sensor using stochastic approach,” in IEEE ISR, 2021.