2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DRL-SFM: Learning Social Navigation from Costmaps and Social Forces for Mobile Robots and Intelligent Wheelchairs

Matthias Kalenberg, Kilian Probst, Andreas Gründer, Christopher May, Jonas Walter and Jörg Franke

***Abstract*— The demand for assistive robots for passenger transport, such as intelligent wheelchairs, is increasing rapidly due to demographic changes. To allow passengers to navigate in crowded environments, such as shopping malls and hospitals, these systems must navigate in a socially accepted manner that ensures the comfort of both passengers and surrounding pedestrians. Although deep reinforcement learning (DRL) has shown promising results for social navigation, existing planners often learn overly passive behaviors, not engaging in the mutual adaptation characteristic of human interaction. In this paper, we introduce a novel DRL-based local planner that learns navigation behaviors by integrating the Social Force Model (SFM) directly into its reward function, allowing more cooperative interactions for mobile robots and intelligent wheelchairs. This approach encourages the agent to learn more forward-looking and reciprocal navigation policies by rewarding actions that align with the dynamics of pedestrians. To ensure generalization and straightforward deployment, our method utilizes the standard Navigation 2 local costmap augmented with pedestrian detections as an observation. The experiments demonstrate that our agent achieves a higher success rate in crowded scenarios with fewer space intrusions, outperforming the state-of-the-art DRL planner based on velocity obstacles by up to 11 %.**

I. Introduction

Social navigation, which enables mobile robots to operate safely and efficiently alongside humans, is an important aspect of current robotics research [1]–[5]. A particularly challenging sub-domain involves mobile robots designed for passenger transport, such as intelligent wheelchairs, which are critical for enhancing user autonomy and reducing caregivers' burden. Unlike delivery or service robots, these platforms introduce a unique constraint: they must ensure passenger comfort while navigating in a socially compliant manner that respects pedestrians' personal space, without disadvantaging the passenger [6]. This is critical for intelligent wheelchairs and inspires novel cooperative navigation strategies for mobile robots. These challenges are substantial in crowded environments, such as hospitals or shopping malls, where achieving safe, predictable, and comfortable navigation remains difficult due to human-robot interactions.

Humans naturally negotiate shared space through mutual adaptation, a process that robots struggle to replicate. However, most robotic navigation frameworks fail to capture reciprocity in social navigation. Although deep reinforcement learning (DRL) planners have demonstrated superior performance over classical methods in crowded environments [7], they often learn overly deferential policies that prioritize passive avoidance over active cooperative negotiation. This can cause the robot to deviate significantly from its path or freeze in place instead of making minor, socially acceptable adjustments to influence pedestrian movement as humans would do. For an intelligent wheelchair, this behavior is particularly problematic, as it consistently disadvantages the passenger by prioritizing all surrounding pedestrians, leading to inefficient and unnatural trajectories.

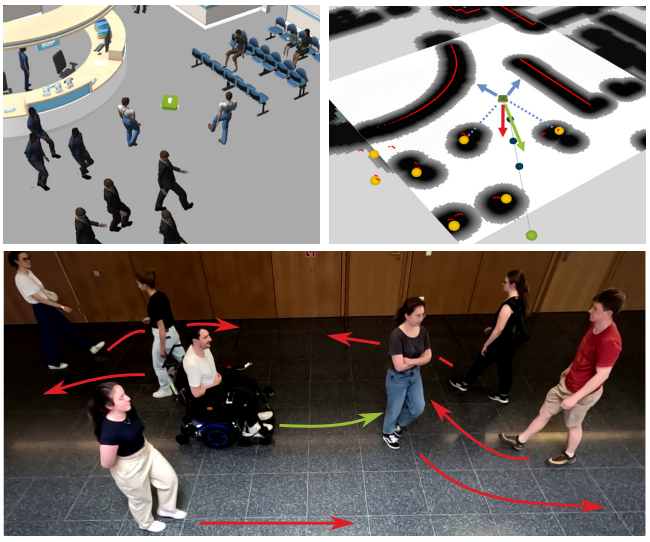


Fig. 1: Learning social navigation from costmaps and social forces. Our deep reinforcement learning planner is trained in simulation (top left), using a novel reward function shaped by the Social Force Model and a costmap-based observation (top right) to learn cooperative, human-like navigation. The resulting policy is evaluated in simulation against state-of-the-art planners and is validated in real-world experiments (bottom) on an intelligent wheelchair.

In this paper, we introduce a DRL-based local planner that integrates the Social Force Model (SFM) [8] into its reward function, allowing a robot to be modeled as an object and an intelligent wheelchair as a person. We hypothesize that by rewarding an agent for actions consistent with social forces, it will learn to navigate more reciprocally and forward-looking, resulting in higher success rates and fewer space intrusions when benchmarking against other planners. The agent's observation space utilizes the local costmap from the standard Navigation 2 framework (Nav2) [9], augmented with pedestrian velocity data. This design not only provides a representation of both static and dynamic obstacles, but also ensures compatibility and facilitates deployment on diverse robotic platforms.

M. Kalenberg, K. Probst, A. Gründer, C. May, J. Walter and J. Franke are with the Institute for Factory Automation and Production Systems (FAPS), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), matthias.kalenberg@faps.fau.de

The main contributions of this work are threefold.

- A novel DRL planner with social-force rewards for social and reciprocal navigation.
- A system architecture that takes advantage of the standard Nav2 costmap, ensuring portability and facilitating deployment across various robotic platforms[2].
- Evaluation in both simulation and real-world experiments with our intelligent wheelchair, demonstrating higher success rates and fewer space intrusions compared to the state of the art.

II. Related Work

Social navigation for mobile robots aims to generate movement paths that are not only collision-free, but also understandable, comfortable, and compliant with social norms when interacting with humans. The increasing focus on this area [1]–[5] has yielded several important insights and challenges. DRL in particular has emerged as a promising method for learning socially acceptable behaviors [4], [5]. At the same time, research results show that the incorporation of mathematical models to predict human behavior can significantly improve robot social interactions [10], [11]. Another challenge is the lack of standardized simulation environments and interfaces, making it difficult to directly compare and benchmark different algorithms [12].

Recent advances in DRL-based local planners demonstrate superior performance in respecting social distances, leading to higher success rates than classical approaches. Initial works focused on decentralized multi-agent collision avoidance, using a value network to enable real-time velocity selection that outperformed classical methods such as Optimal Reciprocal Collision Avoidance (ORCA) in simulated environments [13]. This approach was advanced by incorporating a long short-term memory (LSTM) architecture capable of handling an arbitrary number of dynamic agents [14]. In addition, they removed assumptions about pedestrian behavior, enabling more robust performance in dense scenarios in both simulation and real-world experiments.

While Everett et al. [14] argue for removing assumptions about pedestrian behavior, Xie et al. [7] demonstrate that incorporating velocity obstacle–based assumptions can lead to higher success rates in social navigation. They introduce DRL-VO, a DRL based navigation policy integrating a reward term based on velocity obstacles (VO), which encourages proactive avoidance of pedestrians in crowded environments. Their approach combines a short history of LiDAR scans to capture recent spatial context, kinematic states of nearby pedestrians, and a dynamic subgoal point from a higher-level planner.

Since VO capture only geometric collision avoidance without modeling pedestrian behavior, models such as ORCA [15] and SFM [8] provide a better description of pedestrian behavior. ORCA is a decentralized algorithm that enables collision-free navigation for multiple agents, with each agent making minimal adjustments to its speed under the assumption that the other agents are also participating in evasive maneuvers [15]. The SFM estimates trajectories based on forces acting on obstacles and people [8]. However, as these models are approximations that may not reliably predict pedestrian movements or ensure collision-free navigation in every scenario, they are typically incorporated into planning algorithms. Although it has been integrated into classical approaches such as the Dynamic Window Approach (DWA) [16], the time elastic band approach [10], or hybrid approaches [11], its incorporation into DRL has been limited.

Comparability remains a major challenge in social navigation [12], motivating recent approaches to adopt architectures aligned with Nav2 in the Robot Operating System 2 [9], [17]. Güldenring et al. [18] develop a DRL-based local planner that explicitly accounts for human motion, training policies in randomized 2D environments with simulated human interactions, and deploying them as replacement for conventional planners, delivering significant performance gains in a MiR-100 transport robot. Yao et al. [19] propose a map-based DRL approach for crowd-aware navigation that uses both a sensor map of the environment and a pedestrian map encoding human movements, allowing the robot to adapt to pedestrians following various collision avoidance strategies.

The literature discussed above highlights advances in DRL-based social planners; however, several limitations remain that motivate our work. First, while recent approaches show that integrating a mathematical model into the DRL reward is beneficial [7], they have so far been limited to VO. There remains an unexplored opportunity to leverage models that more accurately describe pedestrian behavior, allowing the learning of more forward-looking social navigation policies. Second, a novel planner should adopt a costmap-based approach suitable for simple ROS 2 integration [18], [19]. Finally, all of these approaches focus on a robot that navigates through a crowd of pedestrians and neglect the presence of a passenger of an intelligent wheelchair, which makes mutual interaction even more important.

III. DRL-SFM

This section details our local DRL-SFM planner, designed to enhance social navigation for mobile robots and intelligent wheelchairs. The planner's objective is to improve mutual and predictable interactions while ensuring easy deployment. To this end, we first formulate the problem as a Partially Observable Markov Decision Process (POMDP). We then delineate the core components of our DRL-SFM agent: (1) a multimodal observation space comprising the Nav2 local costmap, pedestrian detections, and waypoints; (2) a convolutional network architecture for processing these spatial data; and (3) our novel reward function, which integrates the SFM to learn socially compliant and cooperative behaviors.

A. Problem Formulation

A POMDP is defined by a tuple $\langle S, A, T, R, \Omega, O, \gamma \rangle$, where S is a set of states, A is a set of actions, T is a set of conditional transition probabilities between states, R is the reward function, Ω is a set of observations, O is a set of

[2]https://github.com/FAU-FAPS/DRL-SFM

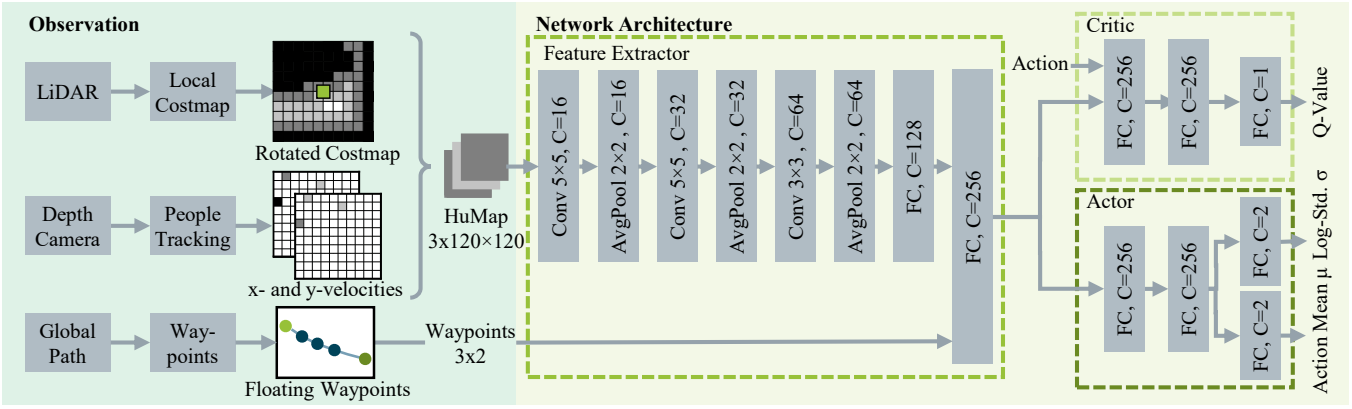


Fig. 2: The architecture of our DRL-SFM local planner. The local costmap and the velocities of the pedestrians, encoded as two additional grids for x- and y-velocities, are stacked to a HuMap, inspired by Xie et al. [7]. It is processed by a LeNet-inspired feature extractor, where C denotes the number of output channels. The features are fused with three floating waypoints representing the global path in polar coordinates relative to the robot. The resulting feature vector serves as the input to the actor and critic. The actor network outputs a probability distribution over all actions. The critic network takes the feature vector and the actor's action as input and outputs a single Q-value estimating the expected future reward.

conditional observation probabilities and γ is the discount factor [20]. This model captures the interaction between a DRL agent and its environment over a sequence of discrete time steps. The agent must learn an optimal policy $\pi^*(a|o)$ to select actions that maximize its cumulative reward. This policy must account for the inherent challenges of social navigation, including the avoidance of collisions with static and dynamic obstacles, as well as mutual social behavior similar to human-human interactions.

B. Observation Space

The observation space Ω is designed to provide the DRL agent with a complete representation of its dynamic environment, consisting of three main components: the local costmap, pedestrian detections, and global path waypoints. To represent the detected pedestrians in the same dimension as the local costmap, we extend the approach of Xie et al. [7]. We represent pedestrians using two grids with the same dimensions as the local costmap: one grid encodes the x-velocity of each pedestrian, and the other encodes the y-velocity. Each pedestrian's velocity is entered into the cell corresponding to their position relative to the robot, allowing the method to handle an arbitrary number of pedestrians. In contrast to their method, we integrate the local costmap of the Nav2 stack into this representation. By matching the dimensions and size of the costmap and pedestrian velocity grids, we stack them into a multi-channel observation, which we introduce as the HuMap. The HuMap then serves as an augmented state representation for the DRL network.

Three floating waypoints, each $1\,\mathrm{m}$ apart, are used to represent the global path, analogously to the local path in Nav2. The global path is updated every $1\,\mathrm{s}$, as is standard in the Nav2 framework. At each time step, we extract three waypoints from the current global path relative to the position and orientation of the agent. These waypoints are encoded as a vector of polar coordinates, which serves as a compact and consistent input to the DRL network. This approach enables the agent to perceive the overall direction of the global path while still allowing it the flexibility to make local adjustments to avoid obstacles and interact socially with pedestrians. Finally, the HuMap is normalized to $[-1.0, 1.0]$, the distances of the agent to the next waypoints to $[0.0, 1.0]$ and the angular deviations to $[-1.0, 1.0]$.

C. Action Space

We use a continuous action space for the local planner, defined by two components: linear velocity v_x and angular velocity ω_z in the robot's local frame. This continuous representation enables the robot to perform smoother and more precise movements. The velocities are constrained to $v_x \in [0.0, 1.0]\,\mathrm{m/s}$ and $\omega_z \in [-1.0, 1.0]\,\mathrm{rad/s}$, allowing the agent to move at human walking speed. Finally, the action space is normalized to $[-1.0, 1.0]$.

D. Network Architecture

Our network architecture is based on a middle-fusion network, designed to efficiently process multimodal sensor data and generate control commands (see Fig. 2). The overall architecture is inspired by the work of Xie et al. [7]. We process our HuMap in a separate feature extractor, after which we connect it with our waypoints. After that, two separate heads for the actor and critic networks follow.

The feature extractor serves as the backbone of our network and is responsible for converting the high-dimensional observation space into a compact set of features. We use a LeNet-inspired [21] architecture to process our multi-channel HuMap. It consists of three convolutional layers, each followed by an average pooling layer, to progressively extract spatial features. All fully connected layers use a rectified linear unit (ReLU) activation function. The output of these layers is then flattened and processed by a fully connected layer. The global path waypoints are passed to the next stage. These two data streams are then concatenated by a fully connected layer, ultimately producing a 256-dimensional feature vector. This design allows the network to

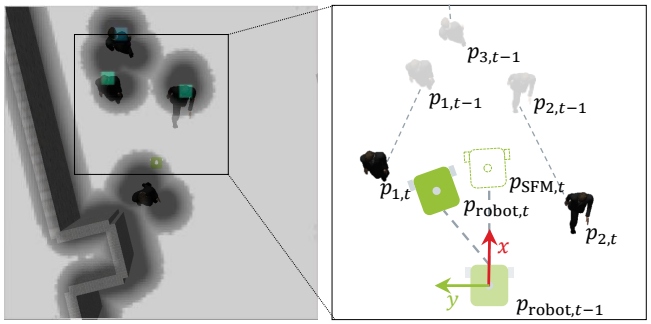


Fig. 3: We introduce a multi-channel observation called HuMap and a novel reward function, $r_{\text{SFM},t}$, based on the SFM to learn socially compliant navigation. (Left) The HuMap encodes static obstacles from the local costmap (gray inflated regions) and the kinematics of nearby pedestrians (cyan). (Right) The reward component $r_{\text{SFM},t}$ penalizes the deviation, $d_{\text{SFM},t}$, between the robot's actual position $\mathbf{p}_{\text{robot},t}$ and a socially-aware position $\mathbf{p}_{\text{SFM},t}$ predicted by the SFM from the previous state $\mathbf{p}_{\text{robot},t-1}$.

integrate the rich spatial information from the HuMap with the goal-oriented guide from the waypoints.

The high-level features extracted from the feature extractor are fed into the separate actor and critic heads. The actor network is responsible for generating the continuous action commands. It consists of two fully connected layers that are then separated into the action mean μ and logarithmic standard deviation σ, from which the action is sampled. The critic network of three fully connected layers is used to estimate the quality of a given state action pair.

E. Reward Function

In addition to standard social navigation rewards, we introduce two novel terms: one from the local costmap and another based on the SFM. The total reward at each time step t is the sum of weighted rewards (see Table I)

$$r_t = r_{\text{goal},t} + r_{\text{trun},t} + r_{\text{prog},t} + r_{\theta,t} + r_{\text{cost},t} + r_{\text{SFM},t}. \tag{1}$$

1) Goal Reward: A reward for reaching the goal and an additional bonus for taking less time. It is described by

$$r_{\text{goal},t} = \begin{cases} w_{\text{goal}} + w_{\text{time}} \frac{t_{\max} - t}{t_{\max} - t_{\min}} & \text{if goal reached} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where w_{goal} is the weight of reaching the goal, w_{time} is the weight of the time bonus. The maximum time is defined as $t_{\max} = 4t_{\min}$. The minimum time $t_{\min}$ is approximated as the time needed for an initial rotation at maximum angular velocity ω_{max} to face the first waypoint and the traversal of the path at maximum linear velocity v_{max}

$$t_{min} = \frac{|\alpha_{\text{waypoint}}|}{\omega_{\max}} + \frac{l_{\text{path}}}{v_{\max}}. \tag{3}$$

2) Truncated Reward: The agent receives a negative reward if the episode is truncated due to a collision or timeout. To reward episodes that have made better progress, the punishment is weighted according to the remaining length of the path l_t. It is formulated as follows:

$$r_{\text{trun},t} = \begin{cases} -w_{\text{trun}} \frac{l_t}{l_{\max}} & \text{if collision} \vee \text{timeout} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where w_{trun} is the weight and $l_{\max}$ maximum path length.

3) Progress Reward: The progress reward encourages the robot to advance toward the next waypoint. We reward progress $d_{\text{prog},t}$ in that direction and punish the agent for moving in the wrong direction. It is formulated as follows:

$$r_{\text{prog},t} = \begin{cases} w_{\text{prog}} d_{\text{prog},t} & \text{if } d_{\text{prog},t} \geq 0 \\ w_{\text{rev}} d_{\text{prog},t} & \text{otherwise,} \end{cases} \tag{5}$$

where w_{prog} weights the positive and w_{rev} negative progress.

4) Heading Reward: To achieve goal-directed behavior, we reward the heading direction as

$$r_{\theta,t} = -w_\theta \frac{|\alpha_{\text{waypoint},t}| - \alpha_\theta}{\alpha_\theta}, \tag{6}$$

where w_θ is the weight, α_{waypoint} is the angle of heading to the waypoint, and α_θ is an angle tolerance.

5) Cost Reward: We introduce a cost reward based on the local costmap to allow the robot to react to costs. Furthermore, this enables minor behavior changes based on the costmap without retraining. However, the evaluation showed that agents with linear cost functions no longer pass through narrow passages. In these situations, low costs should be tolerated, but the higher the costs, the more they should be penalized. Therefore, the following quadratic function has proven to be the most effective

$$r_{\text{cost},t} = -\Big(w_{\text{cost}} \cdot \text{Cost}(\mathbf{p}_{\text{robot},t})\Big)^2, \tag{7}$$

where w_{cost} is the weight and $\text{Cost}(\mathbf{p}_{\text{robot},t})$ the cost in the position of the robot.

6) SFM Reward: The SFM, originally proposed by Helbing and Molnár [8], models the behavior of pedestrians by the sum of an attractive goal force $\boldsymbol{f}_{i,\text{g}}$ and repulsive forces $\boldsymbol{f}_{i,p}$ of other pedestrians $p \in \mathcal{P}$ and $\boldsymbol{f}_{i,o}$ of obstacles $o \in \mathcal{O}$ in the environment. The social force resulting for a person i at time t can be calculated as

$$\boldsymbol{f}_i(t) = \boldsymbol{f}_{i,\text{g}}(t) + \sum_{p \in \mathcal{P}} \boldsymbol{f}_{i,p}(t) + \sum_{o \in \mathcal{O}} \boldsymbol{f}_{i,o}(t). \tag{8}$$

Using the resulting force vector, the movement of a human is simulated. In our implementation, we use the HuNavSim lightsfm library [22], which follows the approach of Moussaid et al. [23], who fitted experimental data of humans into the model parameters.

The SFM reward is formulated as a penalty based on the distance between the robot's actual position $\mathbf{p}_{\text{robot},t}$ after taking an action and the position predicted by the SFM $\mathbf{p}_{\text{SFM},t}$ for the agent derived from the observation at $t-1$. It is formulated as

$$r_{\text{SFM},t} = -w_{\text{SFM}}(\|\mathbf{p}_{\text{robot},t} - \mathbf{p}_{\text{SFM},t}\| - d_{\text{SFM}}), \tag{9}$$

where w_{SFM} is the weight and d_{SFM} is the tolerance for deviation. If the deviation is less than the tolerance, it results in a positive reward (see Fig. 3). For training stability, we normalize the deviation by the duration of the step.

TABLE I: Weights and parameters used for the training.

Parameter	Description	Value
w_{goal}	Goal reached	10
w_{time}	Time bonus	5
w_{prog}	Progress towards waypoint	4.5
w_{rev}	Movement away from waypoint	5.5
w_{trun}	Collision	50
w_{cost}	Cost of local costmap	0.01
w_θ	Weight for heading direction	0.1
α_θ	Tolerated deviation for heading	20°
w_{VO}	Weight for VO	0.6
α_{VO}	Tolerated deviation for VO	30°
w_{SFM}	Weight for SFM	0.3
d_{SFM}	Tolerated deviation for SFM	0.3 m

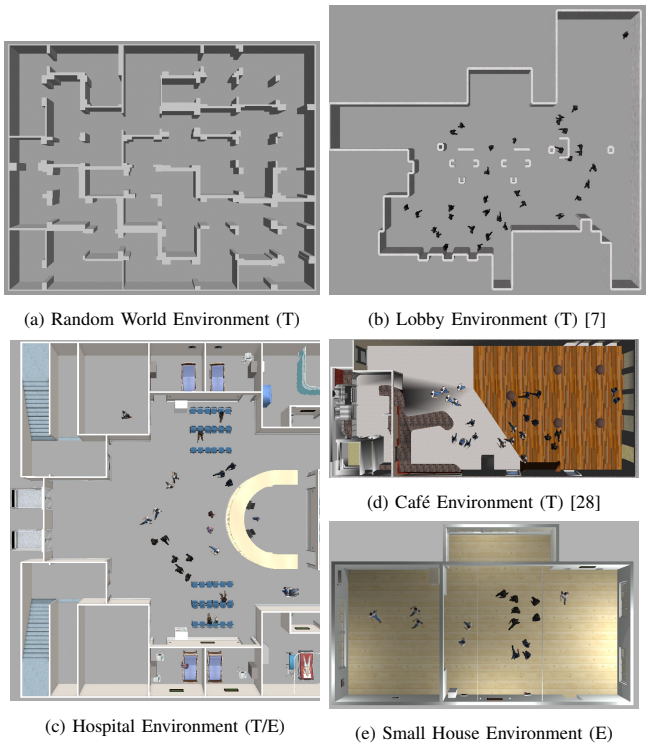
F. Deep Reinforcement Learning Algorithm

We employ the Soft Actor-Critic (SAC) [24] algorithm, an off-policy maximum entropy DRL method, to train our agent. Our choice of SAC is motivated by its demonstrated high sample efficiency and robustness in continuous control tasks [24]. In contrast to on-policy methods like Proximal Policy Optimization (PPO) [25] used by Güldenring et al. [18] and DRL-VO [7], SAC uses a replay buffer to reuse past experiences, significantly reducing the number of interactions with the environment required for convergence. Using Stable Baselines3 [26], our SAC employs two critic networks by default, which improve training stability by reducing positive bias in value estimation. Furthermore, SAC's objective function maximizes both the expected cumulative reward and the policy's entropy, which encourages broader exploration and prevents convergence to suboptimal policies.

G. Simulation and Training

Training is performed in multiple simulated environments that replicate crowded indoor settings with multiple dynamic pedestrian agents. For simulation, we use HuNavSim [22], a framework that enables simulation of multiple pedestrians who perceive the robot as a human in various Gazebo [27] worlds. In total, we train in eight parallel worlds, using four different Gazebo worlds twice each (see Fig. 4). For training, we utilize three realistic scenarios: hospital, cafe, and lobby environments with varying crowd densities. In addition, we use a random world to improve the agent's basic navigation behavior in narrow scenarios. We train the agent using a simulated TurtleBot3 robot model. Training is performed on a workstation equipped with an AMD Ryzen 9 9950X CPU, an NVIDIA RTX 4090 GPU, and 96 GB of RAM.

The agent is trained until the convergence of the policy, evaluated by no increasing mean reward over 50 evaluation episodes. We use the Stable Baselines3 SAC algorithm [26] with an Adam optimizer, using a learning rate η of 0.0003. Training is performed with a replay buffer of 100,000 experiences and a batch size of 512. For policy optimization, we use a discount factor γ of 0.99, a soft update coefficient τ of 0.005 for the target networks, and a target entropy of -2.0 to balance exploration and exploitation. Training takes 22 h with 1,249,280 steps in all environments, after which we select the model with the highest reward.



(a) Random World Environment (T) (b) Lobby Environment (T) [7] (c) Hospital Environment (T/E) (d) Café Environment (T) [28] (e) Small House Environment (E)

Fig. 4: The Gazebo simulation environments used for training (T) and evaluation (E). Training is conducted in the random world, lobby, the hospital, and the café (a-d). Evaluation is done in the known hospital and the unknown small house (c, e).

IV. Experiments

For evaluation, we conduct benchmarks in simulation and real-world experiments. The section is structured as follows: first, we describe the simulation setup, second, we analyze success rates and space intrusions, and finally, we present real-world experiments with our intelligent wheelchair.

A. Experimental Setup in Simulation

To ensure an objective comparison of different local planners, we implement a reproducible evaluation procedure within the simulation. For each scenario, a set of 100 navigation tasks is generated that have a minimum length of 7 m and are guaranteed to go through a crowd. For each planner, we use the same trajectories to ensure that every planner has the same conditions. We evaluate them in a known environment and an unknown environment.

We benchmark our DRL-SFM planner against two deterministic planners and a state-of-the-art reinforcement learning approach. DWA [29] is selected for its widespread use, enabling comparison with other studies. However, the Model Predictive Path Integral Controller (MPPI) [30] is proposed as the new local planner in Nav2 due to its proficiency in dynamic environments. For comparison with a learning-based approach, we use DRL-VO [7]. We evaluate both the original model and a retrained agent in our simulation environment to enable a consistent comparison. The retrained DRL-VO model performed better in our evaluation. This is

TABLE II: Quantitative comparison showing the performance of our proposed DRL-SFM planner in 100 trials against baseline methods in known and unknown environments with five, ten, and 15 pedestrians (Ped.). Metrics include success rate (Succ.), collision rates with persons (Pers.) and obstacles (Obst.), and timeouts (Time.) in %.

		Known Hospital				Unknown Small House			
Ped.	Method	Succ.	Pers.	Obst.	Time.	Succ.	Pers.	Obst.	Time.
5	DWA	84	14	0	2	83	15	2	0
	MPPI	89	11	0	0	91	8	1	0
	DRL-VO	83	5	12	0	78	18	4	0
	Cost2	88	12	0	0	87	12	1	0
	Cost2-VO	86	13	1	0	88	10	2	0
	DRL-SFM	**97**	3	0	0	**93**	7	0	0
10	DWA	76	22	0	2	72	25	3	0
	MPPI	90	10	0	0	77	22	1	0
	DRL-VO	79	8	12	1	52	43	5	0
	Cost2	88	9	3	0	74	24	2	0
	Cost2-VO	84	16	0	0	75	22	3	0
	DRL-SFM	**95**	5	0	0	**81**	18	1	0
15	DWA	58	41	0	1	55	40	5	0
	MPPI	75	25	0	0	63	35	2	0
	DRL-VO	58	30	12	0	42	51	7	0
	Cost2	76	22	2	0	61	38	1	0
	Cost2-VO	77	23	0	0	68	30	2	0
	DRL-SFM	**79**	21	0	0	**75**	25	0	0

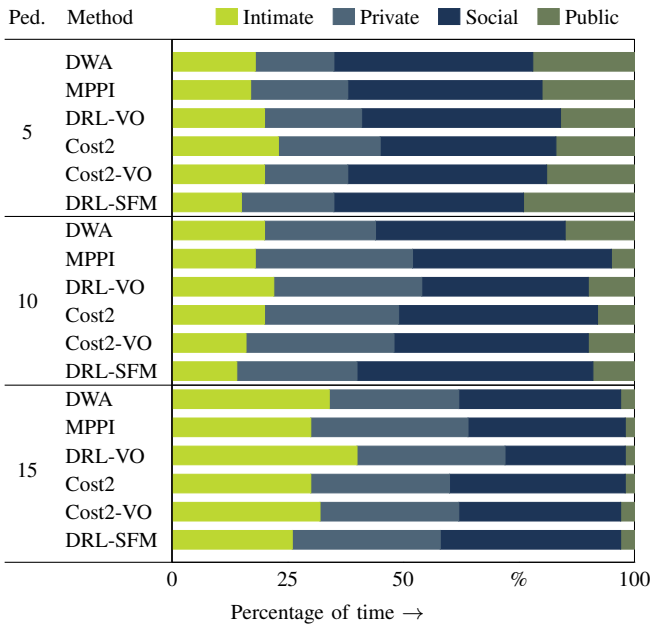


Fig. 5: Time percentages of intrusions of the robot in the intimate ($< 0.45\,\mathrm{m}$), private ($< 1.2\,\mathrm{m}$), social ($< 3.6\,\mathrm{m}$), and public space [31], of other pedestrian (Ped.). The experiments show that our approach achieves less intrusion into people's social spheres.

due to differences in our simulation, such as the fact that our humans move their legs.

We evaluate our cost-based observation and the differences between the VO and SFM rewards. Therefore, we train different models, all using the rewards r_{goal}, r_{trun}, r_{prog}, and r_{θ}, but adding different additional rewards. First, we train the Cost2 model using our observations and the quadratic cost reward r_{cost}. Second, we train the Cost2-VO agent using the cost reward and the VO reward, as described by Xie et al. [7]. Third, the DRL-SFM agent incorporates both the cost reward r_{cost} and the SFM reward r_{SFM}.

B. Simulation Results

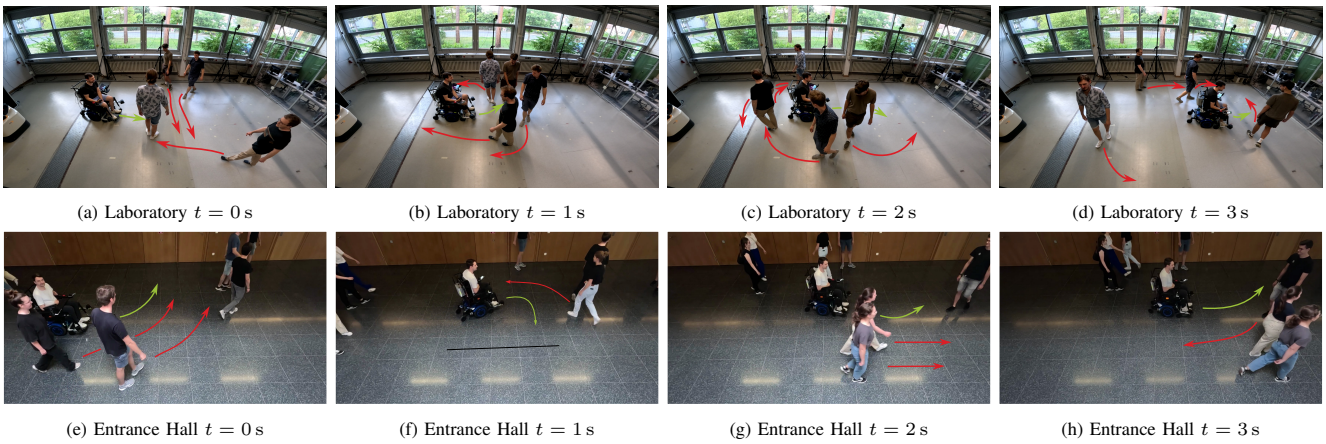
In general, our DRL-SFM shows the highest success rate in our evaluation with an improved success rate of up to $20\,\%$ compared to the DWA (see Table II). However, in situations that involve only five participants, the success rates of DWA and the learning-based methods are very similar. With an increasing number of pedestrians, the success rates of DWA and MPPI decrease. However, it should be noted that MPPI performs significantly better than DWA. The evaluation shows that the DRL methods perform better in more crowded environments. In the following, the effects of our different modules are analyzed in more detail.

The results show that we were able to reproduce the improved success rates of the VO approach with our retrained Cost2-VO agent, compared to both Cost2 without VO and DWA. The Cost2 agent using only the HuMap without any additional reward also performs better than DWA in some dynamic scenarios, such as the hospital with 15 people. This improvement depends on the environment; for example, in the small house with 10 people, the success rates differ only by $2\,\%$. Integrating VO into the reward provides advantages in dynamic scenarios, yielding higher success rates than DWA in all experiments. Our scenarios seem to be more challenging, as is evident by the lower success rate of DWA compared to Xie et al. [7]. However, the costmap representation consistently results in fewer collisions with static objects, particularly in narrow passages or doorways.

Our experimental results show that the DRL-SFM agent trained with the SFM reward achieves higher success rates in all experiments compared to the VO approach (Cost2-VO). Specifically, in the small house scenario with 15 people, it outperforms Cost2-VO by $7\,\%$, and in the hospital scenario with 10 people by $11\,\%$. These results support our hypothesis that the SFM-based reward promotes more forward-looking and socially aware navigation. In contrast, the VO approach relies only on the current relative speed of pedestrians and does not consider additional influences from other people or obstacles in the environment.

Furthermore, pedestrian space intrusions are reduced in all experiments (see Fig. 5). Space intrusions measure the percentage of time the robot spends in a human's intimate ($< 0.45\,\mathrm{m}$), private ($< 1.2\,\mathrm{m}$), social ($< 3.6\,\mathrm{m}$), and public space [31], averaged over 100 evaluation runs. This indicates that incorporating the SFM reward leads the agent to more social behavior, spending more time in social space while avoiding private or intimate zones.

Qualitatively, we found that the trained agent learned waiting and interactive behaviors. In situations where no valid action is available, the agent tends to stop and wait, a realistic and socially acceptable behavior also observed by Güldenring et al. [18]. In addition, the agent exhibits interaction behavior by approaching oncoming pedestrians,



(a) Laboratory $t = 0\,\mathrm{s}$ (b) Laboratory $t = 1\,\mathrm{s}$ (c) Laboratory $t = 2\,\mathrm{s}$ (d) Laboratory $t = 3\,\mathrm{s}$

(e) Entrance Hall $t = 0\,\mathrm{s}$ (f) Entrance Hall $t = 1\,\mathrm{s}$ (g) Entrance Hall $t = 2\,\mathrm{s}$ (h) Entrance Hall $t = 3\,\mathrm{s}$

Fig. 6: For validation, we conduct two real-world experiments with our intelligent wheelchair. First, we replicated the evaluation in our lab with one, two, and four pedestrians due to space limitations (top). Second, we evaluated our system in the entrance hall of our university with one, five, and ten pedestrians walking randomly holding an intuitive, not scripted distance to the wheelchair (bottom).

thereby influencing them through its own social force. It is important to note that this behavior is only applicable within the safety radius of the robot. Nevertheless, it opens new possibilities for social navigation and for balancing pedestrian and passenger comfort.

Finally, the results show that despite using a computationally lighter feature extractor, we achieved higher success rates compared to DRL-VO. Our LeNet-inspired feature extractor contains approximately 1.9 million trainable parameters, compared to 2.5 million in the original DRL-VO with its original observation space. This allowed us to perform the training on our workstation as described in Section III-G.

C. Real-World Demonstration

Our intelligent wheelchair is used for real-world evaluation. It is built on an Invacare TDX SP2, which has a differential drive. All processes run on an onboard computer with an Intel Core i9-13900E CPU, an NVIDIA RTX2000 Ada GPU, and $64\,\mathrm{GB}$ RAM. For environmental perception, a Stereolabs ZED2i stereo camera is used, providing visual odometry and pedestrian tracking. In addition, two SICK NanoScan3 LiDAR sensors provide a scan view of $360°$. During deployment, the policy achieves an inference time of less than $10\,\mathrm{ms}$ requiring $412\,\mathrm{MB}$ of GPU memory.

We conduct two experiments, one in our lab and one in the entrance hall of our university (see Fig. 6). In each scenario, we randomly select a target position so that the global path passes through pedestrians. Due to limited space, we conduct experiments with one, two, and up to five pedestrians in our lab. The entrance hall allows us to conduct experiments with one, five, and ten pedestrians. The demonstrations are presented in the accompanying video.

Real-world experiments with the intelligent wheelchair successfully validate our simulation-driven approach. The model is deployed directly on the wheelchair without additional fine-tuning. However, due to the differences in the velocity and radius of the TurtleBot3 used in simulation and the real wheelchair, we retrain the model to match the physical parameters of the wheelchair. In all trials, the wheelchair consistently demonstrates social navigation, generating smooth and predictive paths to its goal while maneuvering around pedestrians. The observed behaviors mirror the cooperative strategies learned in the simulation. The pedestrians naturally maintain a comfortable distance from the wheelchair. This suggests that its movements are perceived as predictable and socially acceptable.

D. Discussion

The experimental results demonstrate that adding the SFM to the DRL-based local planner reward function improves the overall success rate. The DRL-SFM agent achieves a higher success rate in all evaluations (see Table II). The enhanced performance compared to an VO-based approach can be attributed to the more precise and anticipatory prediction of pedestrian movements. The SFM takes into account other influences, such as pedestrians and nearby objects. Consequently, the agent learns a more interactive policy than with VO, as it learns that pedestrians can be influenced by others and the agent himself. Consequently, our work can be seen as an extension of the DRL-VO [7].

Furthermore, the use of the local costmap in the observation can result in fewer collisions with static obstacles (see Table II). In our hospital evaluation, the DRL-VO shows a larger number of collisions with static objects than the other approaches. An explanation may be attributed to the selection of waypoints. The DRL-VO pure pursuit algorithm has been observed to select waypoints that are located at greater distances, which can result in collisions occurring at corners or doors. However, the local costmap appears to provide the agent with a better understanding of the surrounding environment, as demonstrated by the Cost2-VO.

V. Conclusion and Future Work

In this paper, we present DRL-SFM, a DRL-based local planner that incorporates social forces into its reward function, allowing a robot to be modeled as an object and an

intelligent wheelchair as a person. We also introduced the HuMap observation space, which combines the costmap with pedestrian detections to ensure compatibility across various robotic platforms. Benchmarks in simulation demonstrated that DRL-SFM achieves up to $11\,\%$ higher success rates compared to the state-of-the-art DRL-VO planner. Finally, we validate the transfer of our policy to an intelligent wheelchair, confirming its effectiveness in handling unscripted human interactions in real-world scenarios.

Although our DRL-SFM planner yields promising results, we identify several limitations to address in future work. First, behavioral diversity can be improved by dynamically randomizing pedestrian goals per episode. Second, the network architecture can be extended to a dual-stream model that processes static and dynamic data through separate feature extractors, using LSTM or transformer architectures to better capture spatio-temporal interactions. Third, our SFM-based reward can be replaced with alternative predictive models, such as ORCA or a learned, data-driven trajectory predictor. Lastly, like other learning-based methods, our policy lacks formal safety guarantees, highlighting the need for a hybrid architecture that integrates the planner with a verifiable safety module for robust real-world deployment.

ACKNOWLEDGMENT

This publication is based on the research project EMGRoll, supported by the Federal Ministry of Research, Technology, and Space (BMFTR) under funding code 16SV9306.

REFERENCES

[1] T. Kruse, AK. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *ROBOTICS AND AUTONOMOUS SYSTEMS*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.

[2] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From Proxemics Theory to Socially-Aware Navigation: A Survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, Apr. 2015.

[3] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Recent trends in social aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 93, pp. 85–104, Jul. 2017.

[4] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core Challenges of Social Robot Navigation: A Survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, Sep. 2023.

[5] P. T. Singamaneni, P. Bachiller-Burgos, L. J. Manso, A. Garrell, A. Sanfeliu, A. Spalanzani, and R. Alami, "A survey on socially aware robot navigation: Taxonomy and future challenges," *The International Journal of Robotics Research*, vol. 43, no. 10, pp. 1533–1572, 2024.

[6] M. Kalenberg, C. Hofmann, S. Martin, and J. Franke, "Human Comfort Factors in People Navigation: Literature Review, Taxonomy and Framework," in *Robotics, Computer Vision and Intelligent Systems*, J. Filipe and J. Röning, Eds. Cham: Springer Nature Switzerland, 2024, vol. 2077, pp. 225–243.

[7] Z. Xie and P. Dames, "DRL-VO: Learning to Navigate Through Crowded Dynamic Scenes Using Velocity Obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, Aug. 2023.

[8] D. Helbing and P. Molnar, "Social Force Model for Pedestrian Dynamics," Jan. 1998.

[9] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, May 2022.

[10] G. Pérez, N. Zapata-Cornejo, P. Bustos, and P. Núñez, "Social Elastic Band with Prediction and Anticipation: Enhancing Real-Time Path Trajectory Optimization for Socially Aware Robot Navigation," *International Journal of Social Robotics*, Apr. 2024.

[11] M. Martini, N. Pérez-Higueras, A. Ostuni, M. Chiaberge, F. Caballero, and L. Merino, "Adaptive Social Force Window Planner with Reinforcement Learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 4816–4822.

[12] Y. Gao and C.-M. Huang, "Evaluation of Socially-Aware Robot Navigation," *Frontiers in Robotics and AI*, vol. 8, p. 721317, 2022.

[13] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Jan. 2017, pp. 285–292.

[14] 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)*. Madrid: IEEE, Oct. 2018, pp. 3052–3059.

[15] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research*, B. Siciliano, O. Khatib, F. Groen, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[16] L. Mozzarelli, M. Corno, and S. M. Savaresi, "Socially Aware Local Planning: A Dynamic Window-based Approach," in *2024 European Control Conference (ECC)*. Stockholm, Sweden: IEEE, Jun. 2024, pp. 3255–3260.

[17] S. Macenski, F. Martin, R. White, and J. G. Clavero, "The Marathon 2: A Navigation System," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 2718–2725.

[18] R. Guldenring, M. Gorner, 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)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 6053–6060.

[19] S. Yao, G. Chen, Q. Qiu, J. Ma, X. Chen, and J. Ji, "Crowd-Aware Robot Navigation for Pedestrians with Multiple Collision Avoidance Strategies via Map-based Deep Reinforcement Learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sep. 2021, pp. 8144–8150.

[20] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000437029800023X

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] N. Pérez-Higueras, R. Otero, F. Caballero, and L. Merino, "HuNavSim: A ROS 2 Human Navigation Simulator for Benchmarking Human-Aware Robot Navigation," 2023.

[23] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proceedings of the Royal Society B: Biological Sciences*, Aug. 2009.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 2018.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017.

[26] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of machine learning research*, vol. 22, no. 268, pp. 1–8, 2021.

[27] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3. Sendai, Japan: IEEE, 2004, pp. 2149–2154.

[28] OpenRobotics. (September) Cafe. Open Robotics. [Online]. Available: https://fuel.gazebosim.org/1.0/OpenRobotics/models/Cafe

[29] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[30] G. Williams, A. Aldrich, and E. A. Theodorou, "Model Predictive Path Integral Control: From Theory to Parallel Computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 344–357, 2017.

[31] E. T. Hall, *The Hidden Dimension*, [1st ed.] ed. Garden City, N.Y.: Doubleday Garden City, N.Y., 1966.