POLITECNICO DI TORINO
Repository ISTITUZIONALE

Adaptive Social Force Window Planner with Reinforcement Learning

Original
Adaptive Social Force Window Planner with Reinforcement Learning / Martini, M., Pérez-Higueras, N., Ostuni, A., Chiaberge, M., Caballero, F., Merino, L.. - (2024), pp. 4816-4822. (2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Abu Dhabi (UAE) 14-18 October 2024) [10.1109/iros58592.2024.10802383].

Availability:
This version is available at: 11583/2996179 since: 2025-01-03T14:20:50Z

Publisher:
IEEE

Published
DOI:10.1109/iros58592.2024.10802383

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

Adaptive Social Force Window Planner with Reinforcement Learning

Mauro Martini[1], Noé Pérez-Higueras[2], Andrea Ostuni[1],
Marcello Chiaberge[1], Fernando Caballero[2], and Luis Merino[2]

***Abstract*— Human-aware navigation is a complex task for mobile robots, requiring an autonomous navigation system capable of achieving efficient path planning together with socially compliant behaviors. Social planners usually add costs or constraints to the objective function, leading to intricate tuning processes or tailoring the solution to the specific social scenario. Machine Learning can enhance planners' versatility and help them learn complex social behaviors from data. This work proposes an adaptive social planner, using a Deep Reinforcement Learning agent to dynamically adjust the weighting parameters of the cost function used to evaluate trajectories. The resulting planner combines the robustness of the classic Dynamic Window Approach, integrated with a social cost based on the Social Force Model, and the flexibility of learning methods to boost the overall performance on social navigation tasks. Our extensive experimentation on different environments demonstrates the general advantage of the proposed method over static cost planners.**

I. INTRODUCTION

In recent years, service robots have emerged as a promising automation solution in various social contexts, ranging from domestic assistance [1], [2] to health-care [3]. These advancements have opened up new avenues for robotics research, particularly in human-aware navigation. The robotics community has proposed different benchmarks to evaluate the existing social navigation algorithms [4], [5].

However, social navigation is a complex problem that poses contrasting objectives and is often difficult to formulate with an analytical expression, as is usually done in classic navigation cost functions. This complexity arises from the intricate dynamics of human behavior and the multitude of social rules not considered in standard path planning. Different social navigation scenarios have been partially categorized in the literature to build consistent research and benchmarks [6], [7], highlighting unique challenges in each situation. Standard social planners struggle to perform properly in all of them, considering that environmental geometry and features are crucial in constraining navigation in cluttered, narrow passages or wide open spaces. Therefore, the diversity and unpredictability of social scenarios necessitate a more flexible and adaptive approach.

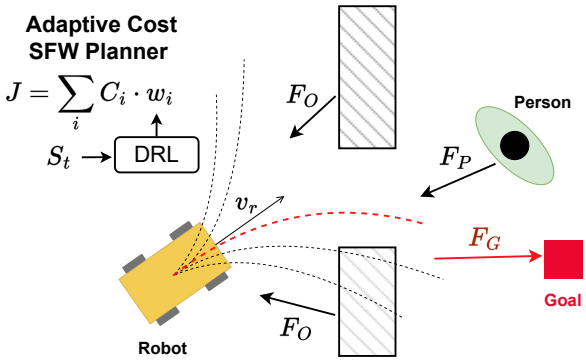


Fig. 1: The Social Force Window (SFW) Planner combines standard Dynamic Window Approach and Social Force Model. The trajectory scoring process is optimized by a DRL agent that dynamically adjust the cost weights based on local environmental conditions.

In this context, Machine Learning (ML) techniques represent a potential solution to this problem. ML models can leverage data to learn behaviors that enhance mobile robots' adaptability to new situations[8] without being explicitly programmed for a specific task. Among existing ML paradigms, Deep Reinforcement Learning (DRL) is particularly suited for learning behavioral policies and, hence, for navigation [9]. Recent studies tried to address the challenges posed by human-aware navigation with DRL [10], as better discussed in Section II. On the other hand, end-to-end learning approaches may often present less robust performance and poor generalization to unseen testing conditions. The authors in [11] have proposed a precious comparison between end-to-end and parameter-learning approaches, highlighting the improved performance of hybrid solutions combining standard controllers and learning.

This work proposes an adaptive parameter-learning approach for social navigation. A social controller is designed by adding a social cost to the Dynamic Window Approach (DWA). This cost is computed considering the robot-pedestrian interaction according to the Social Force Model (SFM) [12], [13]. The proposed solution leverages the advantage of a DRL agent to dynamically adapt the cost weights of the human-aware local planner to different social scenarios. . From a general perspective, this research aims to enhance the performance and versatility of service mobile robots in general social contexts. The contribution

This work was partially supported by the projects NHoA (PLEC2021-007868) and NORDIC (TED2021-132476B-I00), funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR, and partially by PoliTO Interdepartmental Centre for Service Robotics (PIC4SeR).

[1]Department of Electronics and Telecommunications, Politecnico di Torino, 10129, Torino, Italy. mauro.martini@polito.it, andrea.ostuni@polito.it, marcello.chiaberge@polito.it

[2] School of Engineering, Pablo de Olavide University, Crta. Utrera km 1, Seville, Spain noeperez@upo.es, fcaballero@upo.es, lmercab@upo.es

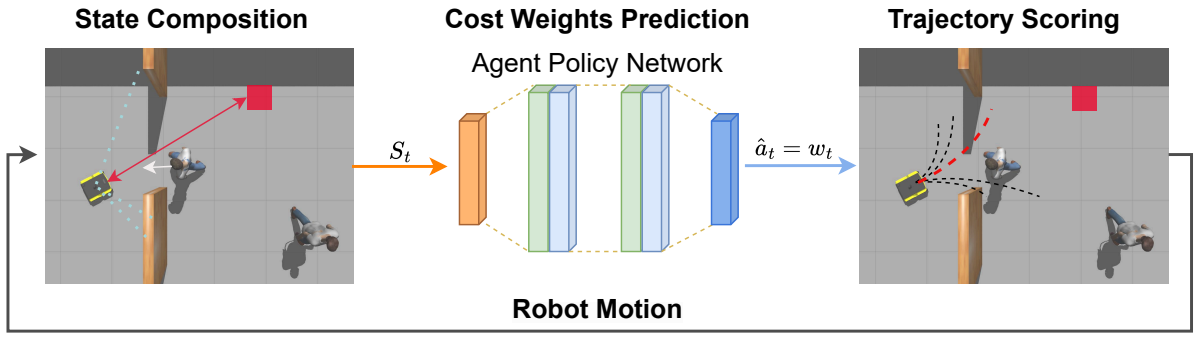


Fig. 2: Workflow of the main step performed by the proposed adaptive Social Force Window (SFW-SAC) Planner with DRL. The policy network learns to set the weights of the social cost used by the DWA for each situation.

of this paper can be summarized in (i) a social-aware local planner based on SFM, used as a baseline solution, that we refer to as Social Force Window (SFW) planner; (ii) an adaptive cost optimization of the SFW planner with a DRL agent, managing the cost terms dynamically according to the context.

II. RELATED WORK

This section presents and discusses the landscape of existing adaptive navigation solutions for general and social scenarios, pointing out Deep learning-driven methods. Diverse end-to-end learning approaches [10], [14] have been directly applied to navigation control. Deep Learning can also select the most suitable social navigation strategy according to the context, as done in [15], which provides the robot with adaptive behavior.

Inverse Reinforcement Learning (IRL) inspired a family of recent works. It is used by [16] to learn diverse cost functions according to the social navigation scenarios. Similarly, the RTIRL [17] and PRTIRL [18] use IRL to adjust the parameters of an RRT* local path planner treated as a black box.

Hybrid solutions between classic controllers and learning methods have been proposed to boost the robustness of autonomous agents. A DRL approach is employed in [19] to evaluate the projected trajectories of the classical Dynamic Windows Approach local planner (DWA), learning a reward function for navigating in dynamic environments. The parameter-learning presented in the family of APPL approaches (Adaptive Planner Parameter Learning) [20] results in a promising direction to convey robustness and versatility in a unique solution [11]. APPL aims to learn a parameter management policy that can dynamically adjust the hyperparameters of classical navigation algorithms according to the environment geometry. The authors proposed different ML techniques, including RL [21]. However, the adaptive parameter approach is applied only in static environments.

Finally, an adaptive DWA implementation has been proposed in [22], dynamically changing the basic cost terms of the algorithm with a Q-table approach. Our work is an improvement and extension of these studies: we frame the adaptive control method in a social navigation problem, adopting a DRL agent working with continuous action space.

III. METHODOLOGY

A. Social Force Window Planner

The Dynamic Window Approach (DWA) is an extremely popular local path planning method in mobile robotics [23]. DWA generates velocity commands that comply with the robot's kinematics constraints. The search space is, therefore, restricted to velocities that can be reached quickly and avoid collisions with obstacles. The classic objective function used to evaluate the trajectories comprises three terms associated with the goal, the velocity, and the obstacles.

A human-aware local planner has been proposed, adding a social cost to the classic DWA trajectory scoring function. For the social cost, a "social work" quantity is adopted by using the Social Force Model (SFM) of interaction between a crowd of agents proposed by [12], [24], [13]. A social work C_s is computed at each time step for the robot according to the following expression:

$$C_s = W_r + \sum_i W_{p,i} \tag{1}$$

where W_r is the sum of the modulus of the robot social force (F_P) and the robot obstacle force (F_O) along the trajectory according to the SFM, while W_p is the sum of the modulus of the social forces generated by the robot for each pedestrian i along the trajectory. A schematic representation of forces acting on the robot is shown in Figure 1. The goal produces an attractive force while the obstacles and pedestrians generates different repulsive forces.

The overall cost function for trajectory scoring can be formulated as:

$$J = C_s \cdot w_s + C_o \cdot w_o + C_v \cdot w_v + C_d \cdot w_d + C_h \cdot w_h \tag{2}$$

where we have a single cost term for social navigation C_s, obstacles in the costmap C_o, robot velocity C_v, distance C_d and heading C_h from a local waypoint on a given global path. The costs are combined using weights w that regulate the impact of each term in the velocity command selection. We refer to this advanced social version of the DWA as a Social Force Window (SFW) planner which is publicly available in

Github[1]. This local planner aims to generate safe, efficient, and human-aware paths. However, finding an optimal trade-off between all those desired aspects in every environmental context is not easy. Hence, we tackle the challenge by using a Reinforcement Learning approach to dynamically handle the weights of the costs.

B. Deep Reinforcement Learning framework

A typical Reinforcement Learning (RL) framework can be formulated as a Markov Decision Process (MDP) described by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$. An agent starts its interaction with the environment in an initial state s_0, drawn from a pre-fixed distribution $p(s_0)$ and then cyclically selects an action $\mathbf{a_t} \in \mathcal{A}$ from a generic state $\mathbf{s_t} \in \mathcal{S}$ to move into a new state $\mathbf{s_{t+1}}$ with the transition probability $\mathcal{P}(\mathbf{s_{t+1}}|\mathbf{s_t}, \mathbf{a_t})$, receiving a reward $r_t = R(\mathbf{s_t}, \mathbf{a_t})$.

In RL, a parametric policy π_θ describes the agent behavior. In the context of autonomous navigation, we usually model the MDP with an episodic structure with maximum time steps T. Hence, the agent's policy is trained to maximize the cumulative expected reward $\mathbb{E}_{\tau\sim\pi}\sum_{t=0}^{T}\gamma^t r_t$ over each episode, where $\gamma \in [0, 1)$ is the discount factor. More in detail, we aim at obtaining the optimal policy π_θ^* with parameters θ through the maximization of the discounted term:

$$\pi_\theta^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi} \sum_{t=0}^{T} \gamma^t [r_t + \alpha\mathcal{H}(\pi(\cdot|s_t))] \tag{3}$$

where $\mathcal{H}(\pi(\cdot|s_t))$ is the entropy term, which increases robustness to noise through exploration, and α is the temperature parameter which regulates the trade-off between reward optimization and policy stochasticity.

In this work, a parameter-learning approach has been adopted to develop an adaptive social navigation system. The DRL agent learns a policy to dynamically set the weights of the cost function that governs the robot's control algorithm. In particular, a Soft Actor-Critic (SAC)[25] off-policy algorithm has been used to train the agent in simulation.

C. SFM Adaptive Cost Approach

The key idea of the proposed method lies in learning an optimal policy to dynamically set the weights of each objective function term used by the SFM local planner to score the simulated circular trajectories and select the next velocity command (v, w). A DRL agent is trained to learn such policy given the local features of the surrounding environment and induce the local planner to choose optimal velocity commands. DRL is considered a competitive approach to tackle this complex task since it is not straightforward for a human to find an optimal trade-off between all the cost terms of a social controller in each situation. On the other hand, the overall methodology represents a robust hybrid solution that efficiently integrates the flexibility of the DRL agent policy with the reliability of a classical navigation algorithm. Moreover, the agent allows the planner to extend its adaptability to different social scenarios by learning the map between task-related and perception data to suitable cost weights. Figure 2 shows the main working steps of the proposed methodology.

[1]https://github.com/robotics-upo/social_force_window_planner

D. Reward function

Reward shaping is a fundamental and controversial practice in model-free RL. A specific reward function, similar to the cost function of the SFW planner, has been designed to let the agent learn an optimal cost weights regulation policy among all the desired components of the overall navigation behavior.

Goal distance First, a distance advancement reward term is defined to encourage the approach of the next local goal on the global path, always placed at $2m$ from the robot's actual pose:

$$r_d = d_{t-1} - d_t \tag{4}$$

where d_{t-1} and d_t are Euclidean distances between the robot and the local goal. Local goal and final goal coincide in the final section of the trajectory.

Path alignment Then, we define a reward contribution r_h to keep the robot oriented towards the next local goal:

$$r_h = \left(1 - 2\sqrt{\left|\frac{\phi}{\pi}\right|}\right) \tag{5}$$

where ϕ is the heading angle of the robot, namely the angle between its linear velocity and local goal on the global path.

Robot velocity A velocity reward is defined to promote faster motion when allowed by the environment:

$$r_v = \frac{v - v_{max}}{v_{max}} \tag{6}$$

Obstacle avoidance An obstacle reward is included to encourage safe trajectory scoring and avoid collisions:

$$r_o = \frac{d_{o,min} - lidar_{max}}{lidar_{max}} \tag{7}$$

where $d_{o,min}$ is the lowest distance measured by the LiDAR ranges at the current time step and $lidar_{max}$ is the saturation distance of LiDAR points, which is set to $3m$ to perceive only local environmental features.

Social penalty The main reward contribution has been assigned to provide the agent with a socially compliant navigation policy. In particular, two different social terms have been considered: proxemics-based reward and social work. The proxemics term penalizes the robot when intruding into the personal space of a person:

$$r_p = \frac{1}{d_{p,min}} \tag{8}$$

where $d_{p,min}$ is the minimum person distance from the robot. The social work generated by robot and people interaction according to the Social Force Model has been used as reward r_s, as done in the cost of the SFW planner.

We also include a reward contribution for end-of-episode states, assigning $r_c = -400$ if a collision occurs. The final

reward signal is finally obtained linearly, combining the described terms. More in detail, $c_d = 10.0$, $c_h = 0.4$, $c_o = 2.0$, $c_p = 2.0$ and $c_s = 2.5$ are the numerical coefficients chosen to balance the diverse reward contributions in the final signal.

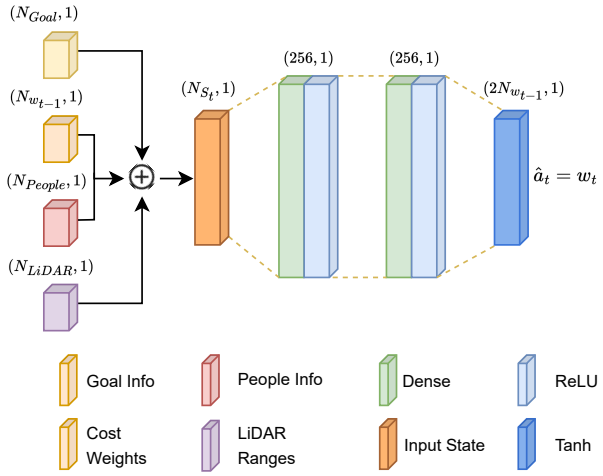


Fig. 3: Schematic of the policy network architecture. State composition is illustrated with separate inputs: goal distance and angle, previous cost weights, people position and velocity, and LiDAR ranges. The new cost weights are predicted as output action of the policy network.

E. Policy Neural Network and Training Design

We define the parametrized agent policy with a deep neural network. We train the agent with the Soft Actor-Critic (SAC) algorithm presented in [25], which allows for a continuous action space and a fast convergence. In particular, we instantiate a stochastic Gaussian policy for the actor and two Q-networks for the critics. The neural network architecture of the agent, represented in Figure 3, is composed of two dense layers of 256 neurons each. A random initial exploration phase has been performed. Random actions are then sampled with a probability that is exponentially reduced with the increase of episodes to maintain a proper rate of exploration during the whole training. Moreover, SAC uses a stochastic Gaussian policy that outputs the mean and the standard deviation of each action distribution, which are used to sample the action value at the training phase. Differently, the mean value of actions' distribution is directly used at test time. The critic networks' structure presents no differences, except they include the predicted action vector in the inputs and predict the Q values.

1) State definition: The information included in the input state of the policy network has been selected to be a synthetic but complete description of the main environmental and task-specific aspects. The state s_t has been, therefore designed as the ensemble of:

- Goal information: $[goal_{angle}, goal_{distance}]$ with respect to the robot expressed in polar coordinate.
- The set of cost weights predicted at the previous time step, $[w_d, w_h, w_v, w_o, w_s]_{t-1}$ to provide information about the actual state of the SFW costs used for trajectory scoring.
- People position and velocity information is embedded in the state for the closest $K = 4$ people to the robot. Position is computed in polar coordinates $[person_{angle}, person_{distance}]$, while velocity with module and orientation, both expressed in the robot frame. People are perceived at a maximum distance of $5m$, and if people are detected to be less than $K = 4$, padding at the maximum distance is used to fill the empty input features and guarantee a constant input dimension.
- A set of 36 LiDAR 2D points saturated at $3m$ to provide the agent with the necessary awareness of local environmental geometry and spaces and the presence of obstacles.

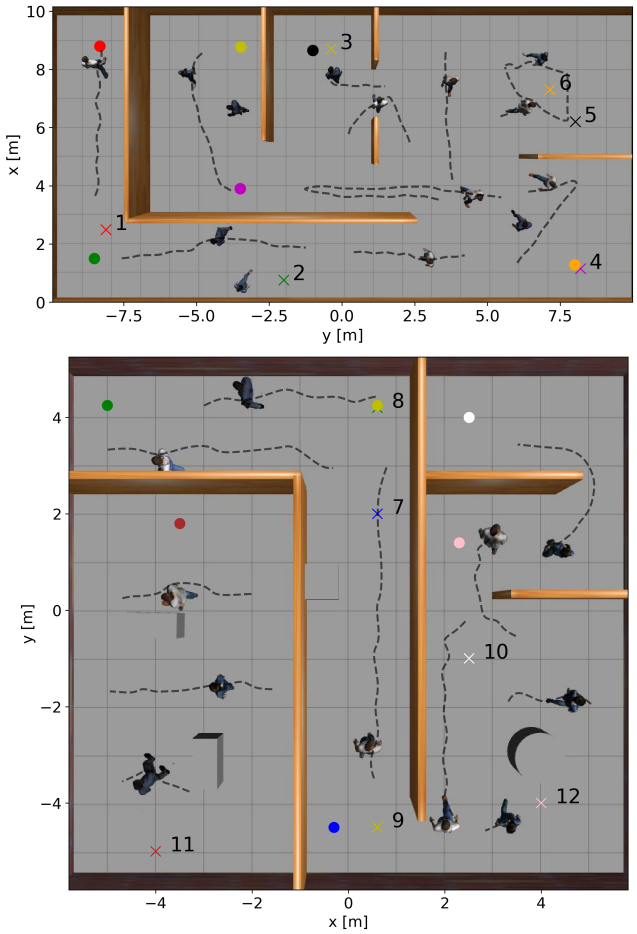


Fig. 4: Gazebo simulation environments where the agent has been trained (top) and tested (top and bottom). People trajectories are indicated with dotted lines, robot starting poses with a circle, goals with a cross and the associated episode's number.

2) Output actions: The policy network predicts an action $a_t = [w_d, w_h, w_v, w_o, w_s]$ at each time step, directly representing the new set of cost weights for the Social Force

Window local planner. The weights are chosen in the interval of values $[0.1, 5.0]$, and they are set with a frequency of $2Hz$, which has been considered a proper choice for a robot moving at a maximum translational velocity of $v_{max} = 0.6[m/s]$ in dynamic social scenarios.

IV. EXPERIMENTS AND RESULTS

A. Experimental settings

The adaptive social navigation system has been trained and tested in Gazebo simulation in diverse scenarios. The HuNavSim plugin [5] has been adopted to instantiate people moving according to the SFM with customized trajectories and behaviors; HuNavSim has also been used to collect the metrics of interest for the evaluation of the algorithms. Differently, the PIC4rl-gym [26] has been used as ROS 2 framework for DRL agent training and testing. Figure 4 shows the environments realized to carry out challenging experiments categorized in different social challenges. The first Gazebo world is used for both training and testing. A wide set of diverse episodes is defined for training the agent in various conditions involving pedestrians passing, overtaking, and crossing tasks in narrow and open spaces. The agent has also been partially tested in the same world, changing the starting pose of the robot and its goals, indicated in Figure 4, scenarios $[1-6]$. Diversely, testing episodes $[7-12]$ have been performed in a separate different world to evaluate the system in diverse scenarios, always considering crossing, passing, overtaking, and mixed miscellaneous tasks.

For general and reproducible experimentation, we set a basic pedestrian behavior that considers the robot an obstacle. A global path is computed once at the beginning of each episode with the standard grid-based search planner of the Nav2 framework. The main controller parameters of the SFW algorithm are the ones of the classic DWA. Besides the kinematics limits of the robots, the waypoint position and the trajectory simulation time are important factors for a social controller, regulating the alignment to the global path and the predicting horizon. Controller parameters and cost weight values used for the experimentation are reported in Table I. The static cost weights used are the results of the fine-tuning process carried out by a human expert. We use the same implementation of the SFW planner for the DWA baseline, setting the social cost to zero value.

TABLE I: Controller parameters. On the left the kinematic and classic DWA parameters, on the right the cost function weights used by the SFW controller and modified by the SFW-SAC in the range reported. The DWA uses the same cost weights except for the social term.

DWA parameter	Value	Cost weight	SFW	SFW-SAC
max linear vel	0.6 [m/s]	social weight	2.0	[0.5 - 3.0]
min linear vel	0.08 [m/s]	costmap weight	2.0	[0.5 - 3.0]
max angular vel	1.5 [rad/s]	velocity weight	0.8	[0.1 - 1.0]
waypoint tol	2.0 [m]	angle weight	0.6	[0.1 - 1.0]
sim time	2.5 [s]	distance weight	1.0	[0.1 - 1.5]

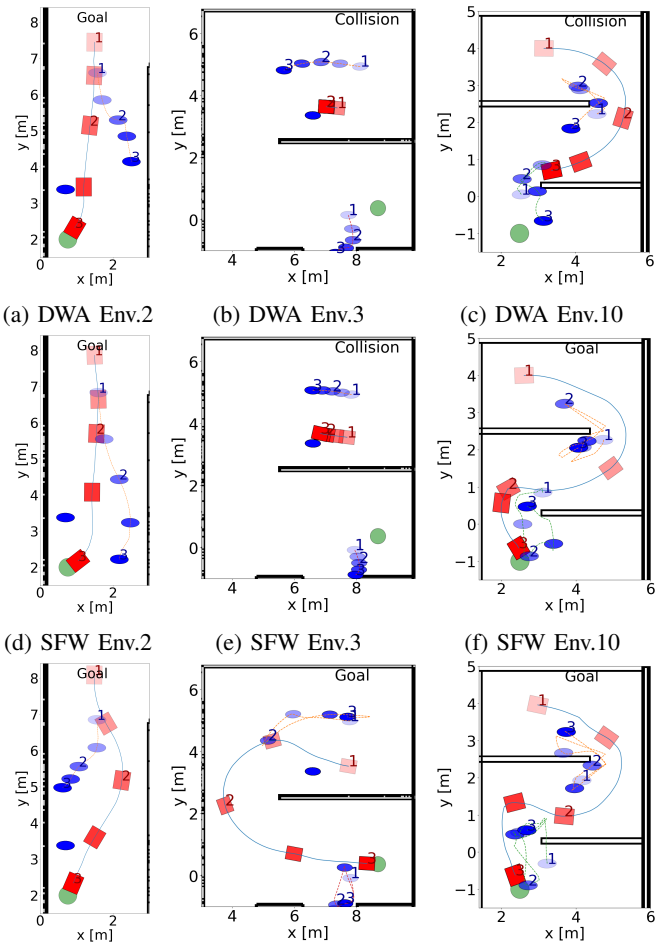
TABLE II: Results obtained testing the adaptive controller SFW-SAC on different environments. We report average metric results over 10 runs, comparing the agent with DWA and SFW baselines.

Env	Method	Success%	Time [s]	PL [m]	v_{avg} [m/s]	SW_{step}
1	DWA	100.00	11.65	5.68	0.49	0.03
	SFW	100.00	12.48	6.04	0.49	0.04
	SFW-SAC	100.00	12.21	5.91	0.48	0.05
2	DWA	20.00	12.32	6.21	0.50	0.17
	SFW	70.00	20.43	6.42	0.31	0.11
	SFW-SAC	100.00	13.08	6.36	0.49	0.22
3	DWA	0.0	-	-	-	-
	SFW	0.0	-	-	-	-
	SFW-SAC	70.00	23.26	11.29	0.48	0.12
4	DWA	20.00	21.71	12.21	0.56	0.25
	SFW	60.00	37.91	12.65	0.38	0.16
	SFW-SAC	70.00	26.20	12.60	0.48	0.20
5	DWA	50.00	19.53	9.47	0.49	0.28
	SFW	60.00	29.34	9.54	0.34	0.34
	SFW-SAC	70.00	33.28	9.47	0.30	0.24
6	DWA	50.00	19.57	8.87	0.46	0.26
	SFW	40.00	44.24	10.75	0.25	0.16
	SFW-SAC	90.00	23.60	8.75	0.37	0.18
7	DWA	0.0	-	-	-	-
	SFW	90.00	19.83	6.38	0.32	0.08
	SFW-SAC	100.00	16.59	6.25	0.39	0.11
8	DWA	90.00	10.35	5.21	0.50	0.10
	SFW	100.00	15.64	5.95	0.35	0.13
	SFW-SAC	100.00	12.32	5.42	0.44	0.16
9	DWA	80.00	15.75	8.32	0.53	0.14
	SFW	100.00	19.77	8.84	0.45	0.12
	SFW-SAC	100.00	18.62	8.98	0.48	0.12
10	DWA	0.0	-	-	-	-
	SFW	70.00	31.96	8.91	0.29	0.13
	SFW-SAC	90.00	32.84	9.99	0.29	0.14
11	DWA	50.00	15.45	6.98	0.45	0.29
	SFW	80.00	48.58	8.37	0.19	0.16
	SFW-SAC	80.00	30.60	7.72	0.27	0.17
12	DWA	90.00	13.98	6.09	0.43	0.21
	SFW	40.00	53.04	6.48	0.14	0.16
	SFW-SAC	90.00	32.01	6.29	0.24	0.20
Avg	DWA	58.57	15.84	8.00	0.50	0.17
	SFW	76.67	25.73	8.39	0.35	0.14
	SFW-SAC	89.00	21.20	8.50	0.42	0.15

B. Results

In this section, we describe the metrics chosen to evaluate the proposed social navigation system, and we discuss the obtained results. Considering the difficulty of strictly judging the performance of a social navigation algorithm without adopting human rating, the adaptive social planner SFW-SAC has been analyzed from different perspectives. First, we compared it to the baselines DWA and Social Force Window Planner (SFW) with static costs using relevant quantitative metrics.

Quantitative evaluation Standard navigation metrics such as clearance time $[s]$, path length (PL) $[m]$ and average linear velocity $v_{avg}[m/s]$ are employed to evaluate the effectiveness of the planner from a classic perspective. On the other hand, the social work (SW) metric is included in the quantitative results to show the social impact of the navigation, measuring the social forces generated by the robot and on the robot by pedestrians during its motion. The social work has been taken into account as the average value SW_{step} over the number of trajectory steps to consider the duration of the episode, and avoiding metrics biases caused by a fast execution of the navigation task.



(a) DWA Env.2 (b) DWA Env.3 (c) DWA Env.10

(d) SFW Env.2 (e) SFW Env.3 (f) SFW Env.10

(g) SFW-SAC Env.2 (h) SFW-SAC Env.3 (i) SFW-SAC Env.10

Fig. 5: Trajectory plots of Env 2, 3, 10 comparing DWA, SFW and the proposed SFW-SAC adaptive planner with DRL. Goals are represented with green circles, the robot with a red rectangle and people with blue ellipses. Transparency and indexes 1,2,3 represent temporal evolution of the motion of both robot and people.

A thorough inspection of the performance is presented, reporting both resulting metrics in Table II. Results show that the DRL method enables the planner to overcome the baseline performance in multiple environments. The basic DWA fails to complete the navigation task in a high percentage of scenarios, colliding with obstacles or pedestrians. On the other hand, the SFW baseline demonstrates an improved ability to handle social navigation tasks. Even though the cost weights combination found by a human offers safe behavior in most situations, it still presents some limitations. For example, in cluttered scenarios, SFW can be hindered by high social costs, which can cause the algorithm to get stuck. Diversely, the SFW-SAC proposes a more general performance, finding a better trade-off of costs in different situations. This advantage is proved by the higher success rate obtained in almost all the environments, sometimes being the unique solution able to complete it (Env 3). A more detailed analysis of results tells us that SFW-SAC often finds a compromise performance between the more aggressive DWA and the SFW with high static social cost values. This trend can be noticed by looking at the time, path length, and average velocity results. On average, the adaptive planner generally chooses higher velocities than the SFW but lower than the DWA. Social work embeds all the navigation effects on humans, considering distances, approaching velocities, and time spent close to people. Thus, it often presents alternating results that are difficult to interpret without a visual inspection of the navigation. Indeed, DWA often reduces the duration of the episode thanks to abrupt motion and brief transitions close to people that can lead to a collision with a high risk. The SW_{step}, relating the social impact to the duration of the task, shows more clearly the socially compliance of SFW and SFW-SAC compared to DWA. The agent-based solution is often able to mitigate the social work improving or remaining comparable with the SFW, without compromising the success rate or strongly violating social rules.

Proxemics According to this, the human awareness of navigation is also measured through the level of intrusion of proxemics spaces of people. Figure 6 illustrates the percentage of time spent by the robot in the intimate, personal, social, and public space of people in each testing episode. It can be noticed that even though the SFW-SAC planner develops a more risky navigation policy, it can keep people's distances respected and comparable with the SFW baseline. It should be noted that the proxemics results reported should be paired with the success rate of the algorithms on each episode for a clear perspective. DWA often computes aggressive trajectories that do not take humans into account, although the temporal intrusion of social spaces is sometimes limited to short time intervals.

Trajectory visual comparison Resulting trajectories of some relevant scenarios where the proposed adaptive SFW-SAC show significant improvements and interesting differences, i.e., Environments $[2, 3, 10]$, are plotted in Figure 5 for a better understanding of the performance of the algorithms. In Env. 2, only the SFW-SAC can properly perform the overtaking, passing to the left of the person, while the other algorithms cross the person's path. In Env. 3, DWA and SFW are not able to handle the presence of a static person on the path and collide; the agent learns how to deviate the motion from the path and avoid the person. In Env. 10, a narrow curved passage with people passing is successfully handled by the SFW and the SFW-SAC, with a smoother trajectory.

V. CONCLUSIONS AND FUTURE WORK

In this work, an adaptive social planner is proposed, combining a social DWA approach with a Deep Reinforcement Learning agent. The agent boosts the performance and versatility of social navigation, learning how to adapt the controller's cost terms weights to environmental context-specific conditions. The results obtained show an improvement in both success rate and navigation metrics, balancing the trade-off between standard navigation effectiveness and

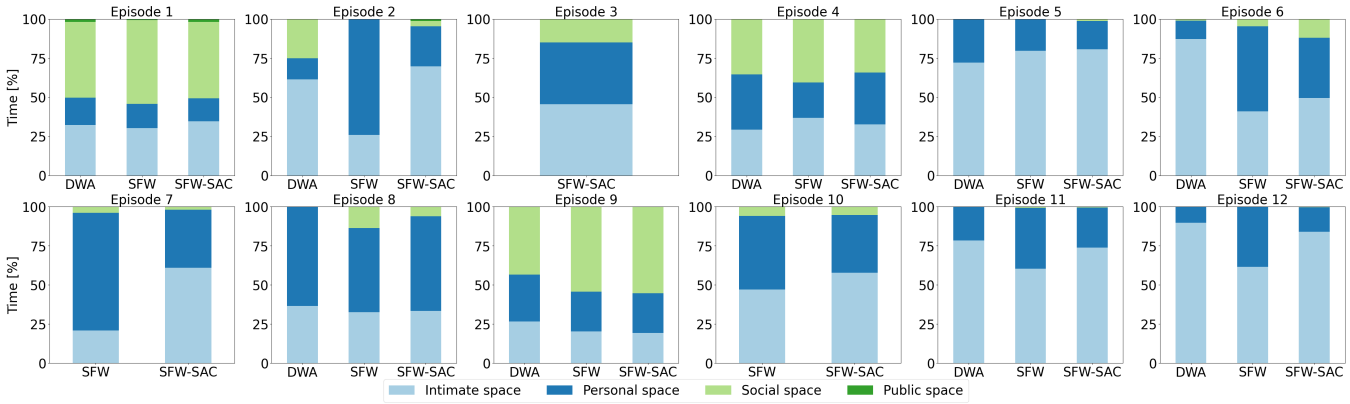


Fig. 6: Average temporal percentage of pedestrians space intrusion according to the proxemics standard in 10 different scenarios. Proxemics data must be coupled with success rate reported in Table II for a clear performance frame.

social rules. Some relevant trajectories are also visualized for a clearer evaluation of the algorithms. Future works will see the enrichment of the experimentation in different scenarios and testing the system on the real robot, including a perception system to estimate the necessary state information of the closest people to the robot. Furthermore, the proposed method can be extended to other social controllers to be included in a common benchmark and, finally, to learning adaptive behaviors from a global planning perspective.

REFERENCES

[1] A. Eirale, M. Martini, and M. Chiaberge, "Human-centered navigation and person-following with omnidirectional robot for indoor assistance and monitoring," *Robotics*, vol. 11, no. 5, p. 108, 2022.

[2] A. Eirale, M. Martini, L. Tagliavini, D. Gandini, M. Chiaberge, and G. Quaglia, "Marvin: An innovative omni-directional robotic assistant for domestic environments," *Sensors*, vol. 22, no. 14, p. 5261, 2022.

[3] J. Holland, L. Kingston, C. McCarthy, E. Armstrong, P. O'Dwyer, F. Merz, and M. McConnell, "Service robots in the healthcare sector," *Robotics*, vol. 10, no. 1, p. 47, 2021.

[4] H. Khambhaita and R. Alami, "Assessing the social criteria for human-robot collaborative navigation: A comparison of human-aware navigation planners," in *2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN)*. IEEE, 2017, pp. 1140–1145.

[5] N. Pérez-Higueras, R. Otero, F. Caballero, and L. Merino, "Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7130–7137, September 2023.

[6] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Frontiers in Robotics and AI*, vol. 8, p. 721317, 2022.

[7] 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, 2023.

[8] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion control for mobile robot navigation using machine learning: a survey," *arXiv preprint arXiv:2011.13112*, 2020.

[9] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021.

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

[11] Z. Xu, X. Xiao, G. Warnell, A. Nair, and P. Stone, "Machine learning methods for local motion planning: A study of end-to-end vs. parameter learning," in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2021, pp. 217–222.

[12] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[13] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS one*, vol. 5, no. 4, p. e10047, 2010.

[14] R. Mirsky, X. Xiao, J. Hart, and P. Stone, "Prevention and resolution of conflicts in social navigation–a survey," *arXiv preprint arXiv:2106.12113*, 2021.

[15] S. B. Banisetty, V. Rajamohan, F. Vega, and D. Feil-Seifer, "A deep learning approach to multi-context socially-aware navigation," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 23–30.

[16] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *International Journal of Social Robotics*, vol. 8, no. 1, pp. 51–66, 2016.

[17] N. Perez-Higueras, F. Caballero, and L. Merino, "Teaching Robot Navigation Behaviors to Optimal RRT Planners," *International Journal of Social Robotics*, vol. 10, no. 2, pp. 235–249, 2018.

[18] Z. Ding, J. Liu, W. Chi, J. Wang, G. Chen, and L. Sun, "Prtirl based socially adaptive path planning for mobile robots," *International Journal of Social Robotics*, 2022.

[19] U. Patel, N. Kumar, A. J. Sathyamoorthy, and D. Manocha, "Dynamically feasible deep reinforcement learning policy for robot navigation in dense mobile crowds," 2020. [Online]. Available: https://arxiv.org/abs/2010.14838

[20] X. Xiao, Z. Wang, Z. Xu, B. Liu, G. Warnell, G. Dhamankar, A. Nair, and P. Stone, "Appl: Adaptive planner parameter learning," *Robotics and Autonomous Systems*, vol. 154, p. 104132, 2022.

[21] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, "Applr: Adaptive planner parameter learning from reinforcement," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6086–6092.

[22] M. Kobayashi, H. Zushi, T. Nakamura, and N. Motoi, "Local path planning: Dynamic window approach with q-learning considering congestion environments for mobile robot," *IEEE Access*, vol. 11, pp. 96 733–96 742, 2023.

[23] 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, 1997.

[24] 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*, vol. 276, no. 1668, pp. 2755–2762, 2009.

[25] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[26] M. Martini, A. Eirale, S. Cerrato, and M. Chiaberge, "Pic4rl-gym: a ros2 modular framework for robots autonomous navigation with deep reinforcement learning," in *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*. IEEE, 2023, pp. 198–202.