POLITECNICO DI TORINO
Repository ISTITUZIONALE

A General Social Cost Layer for Robotic Navigation Planning

Original
A General Social Cost Layer for Robotic Navigation Planning / Aisa, F., Ostuni, A., Martini, M., Eirale, A., Leonetti, M., Nazzario, M., Chiaberge, M.. - In: INTERNATIONAL JOURNAL OF SOCIAL ROBOTICS. - ISSN 1875-4805. - ELETTRONICO. - 18:(2026). [10.1007/s12369-026-01384-0]

Availability:
This version is available at: 11583/3009387 since: 2026-03-30T14:09:02Z

Publisher:
Springer

Published
DOI:10.1007/s12369-026-01384-0

Terms of use:

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

Publisher copyright

(Article begins on next page)

18 November 2026

International Journal of Social Robotics (2026) 18:46
https://doi.org/10.1007/s12369-026-01384-0

A General Social Cost Layer for Robotic Navigation Planning

Filippo Aisa[1] · Andrea Ostuni[1] · Mauro Martini[1] · Andrea Eirale[1] · Matteo Leonetti[2] · Matteo Nazzario[3] · Marcello Chiaberge[1]

Received: 8 April 2025 / Revised: 5 January 2026 / Accepted: 13 February 2026
© The Author(s) 2026

Abstract
Recent advancements in indoor robot navigation have driven the increasing presence of service robots in homes and public spaces. While traditional geometric planning methods excel at obstacle avoidance, they often fail to generate human-aware trajectories, limiting their effectiveness in social environments. Social navigation aims to bridge this gap by prioritizing socially acceptable behaviours, ensuring natural human-robot coexistence. Current research predominantly focuses on local control approaches that predict human motion to enhance safety. However, in crowded environments, robots must exhibit human-like behaviour that cannot be effectively addressed through reactive control alone. An integration at the planning level is deemed necessary to prevent disruptions to social dynamics. This study introduces a novel framework that enriches robotic navigation with adaptive social considerations, specifically targeting scenarios where local control methods fall short in maintaining social compliance. Our approach builds upon the robustness of classical grid-based planners while incorporating a learning-based social cost layer. Using an encoder-decoder neural model, we generate a dynamic social cost map from positional data of individuals, environmental geometry, and the robot's goal. This cost map and the static obstacle map are integrated into the planning process, allowing the robot to navigate complex social settings without compromising the default planner functionality. We validate our method across diverse real-world and simulated scenarios, including queuing, group conversations, narrow blind passages, and corridor navigation. The results demonstrate the framework's adaptability, robustness, and ability to generalize to different social contexts, ensuring socially aware robot navigation in dynamic environments.

Keywords Social navigation · Service robotics · Planning · Neural network

1 Introduction

Significant advancements in indoor robot navigation have paved the way for the first generation of commercial service robots, now increasingly present in homes and public venues such as shopping malls, airports, and restaurants [1–3]. These robots primarily rely on standard geometric path-planning methods, simplifying human presence by treating people as dynamic obstacles. However, effective social navigation requires more than just obstacle avoidance: it

✉ Mauro Martini
mauro.martini@polito.it

Filippo Aisa
filippo.aisa@polito.it

Andrea Ostuni
andrea.ostuni@polito.it

Andrea Eirale
andrea.eirale@polito.it

Matteo Leonetti
matteo.leonetti@kcl.ac.uk

Matteo Nazzario
matteo.nazzario@intesasanpaolo.com

Marcello Chiaberge
marcello.chiaberge@polito.it

[1] Department of Electronics and Telecommunications, Politecnico di Torino, C.o Duca degli Abruzzi, Torino 10129, TO, Italy

[2] Department of Informatics, King's College London, Strand, 30 Aldwych, London WC2B 4BG, England, UK

[3] Innovation Center, Intesa San Paolo, C.so Inghilterra, Torino 10138, TO, Italy

Published online: 05 March 2026

demands that robots exhibit socially appropriate behaviours to minimize disruptions and integrate into human environments [4].

Traditionally, research on social navigation has focused on modelling human and crowd motion [5], emphasizing factors such as direction and velocity to help robots respect personal space and respond appropriately in coordination-intensive situations (e.g., navigating narrow corridors) [6]. However, this work argues that there are common scenarios where predictive local control alone is insufficient to ensure socially compliant behaviour. Instead, robots should approach these situations at the planning level first. For example, in scenarios where individuals remain stationary, such as people engaged in group conversations or queuing, robots should exhibit human-like behaviour by proactively avoiding disruptions rather than relying solely on reactive control [7].

To address these challenges, we propose a learning-driven framework that builds upon existing, well-established navigation planners. Our approach prioritizes social awareness in navigation while leveraging the efficiency of conventional obstacle avoidance algorithms. Specifically, our solution combines the robustness of classic grid-based planners with the adaptability of learning-based methods, incorporating social factors that are difficult to define through explicit mathematical cost functions. Unlike end-to-end learning approaches, which often struggle with generalization, our method introduces an additional cost layer tailored to social scenarios, ensuring that social considerations are handled at the planning level while preserving the default planner's behaviour in other contexts. In contrast to prior work that employs classifiers to distinguish between social navigation contexts and apply different control strategies [8], we integrate an adaptive social cost layer that enables consistent behaviour across a broad range of relevant scenarios.

Our approach exploits input data that captures the positions of fixed obstacles, nearby individuals, and the robot's destination. A neural network processes this data to generate a social cost map, which complements the traditional static obstacle map. This novel design effectively transforms the complexities of social navigation into a classical 2D navigation framework, facilitating integration with existing planning architectures.

As a case study, we examine a variety of common public scenarios, including queuing, small group conversations, pedestrian crossings, overtaking, and dynamically evolving environments with occluded entrances. Additionally, we incorporate environmental geometric features to address complex navigation challenges such as corridor traversal and blind corners. While these situations have often been analyzed in isolation, we demonstrate how our approach generalizes across all of them, effectively addressing diverse social settings. Our scalable methodology can be significantly extended to new scenarios by generating corresponding datasets. Experimental validation in both simulated and real-world settings confirms the robot's ability to navigate while respecting social norms. Moreover, it highlights the importance of generalization in achieving effective, socially aware robot navigation across a wide range of environments.

1.1 Paper Organization

The following sections of this article present the proposed research in detail. Section 2 reviews relevant literature on social navigation. Section 3 introduces background concepts, including path planners and cost functions. The overall methodology is described in Sect. 4, covering the dataset generation process, the neural network architecture used to learn the social cost function, and its integration into the navigation system. Section 5 presents and analyzes experimental results from both simulations and real-world tests, evaluating social metrics and trajectory performance. Finally, Sect. 6 discusses conclusions and outlines potential directions for future research.

2 Related Works

Since the advent of autonomous robots, considerable research has been directed toward developing robotic behaviours and planning strategies to account for implicit social rules and enable autonomous platforms to integrate seamlessly into populated environments. Proxemics theory and distance-based approaches have been deeply analyzed [9] to avoid the intrusion of intimate spaces. However, many core challenges are still pending for complete social navigation. A wide range of works adopt Gaussian cost distributions in the direction of human motion to enhance the collision avoidance foresight of robots [10, 11]. The ability to inflate and modify cost functions based on human presence has inspired numerous subsequent works. For instance, Mateus et al. [12] leveraged deep learning combined with hybrid asymmetric Gaussian functions to model pedestrian comfort distances. Similarly, Fang et al. [13] advanced this concept by modelling not only individual humans but also groups, applying adaptive Gaussian cost functions tailored to their positions and configurations.

Other studies have explored an alternative approach by planning around predictions of future human poses. In these cases, a predictor—often based on deep neural networks such as convolutional neural networks [14, 15], generative adversarial networks [16], or attention-based transformers [17]—estimates the motion and future positions of humans

and other dynamic obstacles in the environment. These predictions are then used to generate optimal, obstacle-free, and socially acceptable trajectories toward the robot's goal. Notable examples are the works of Mavrogiannis et al., which introduce geometric [6] and topological invariant [18] representations to model the coupling of trajectories among multiple navigating agents.

In this context, a popular crowd behavioural model is the Social Force Model (SFM) [5], which minimizes the impact the robot causes on surrounding pedestrians. An important advantage of the SFM is its ability to integrate and adapt to different custom situations. For example, it can be tailored in populated environments to achieve natural and human-like movements [19], to follow a dynamic goal [20], to obtain side-by-side navigation [21], and to avoid occluded areas [22].

End-to-end policy learning based on Deep Reinforcement Learning has been extensively employed for prediction in crowd navigation scenarios. Chen et al. [23] introduce CADRL, a DRL framework designed for socially aware multi-agent collision avoidance. Everett et al. [24] adopt an actor-critic variant to relax prior assumptions, enabling simultaneous learning of policies and agent motion models. Additionally, Tai et al. [25] leverage generative adversarial imitation learning, training their model on a dataset generated using the social force model, to enhance socially compliant navigation. Finally, Chen et al. [26] apply attention-based reinforcement learning to develop interaction-aware collision avoidance behaviours.

Most research on social navigation, including the works discussed so far, primarily focuses on people's movement—analyzing their trajectories, intentions, and destinations. However, many social norms are not based on movement but rather on human activity. For example, walking between two people having a discussion is generally considered impolite, even if they are just standing still. Despite their importance, such scenarios are largely under-represented in the scientific literature, with only a few studies addressing them. Some of these studies focus on integrating specific social behaviours into navigation systems. A notable example by Xiao et al. [27] employs a Performer architecture [28] trained using imitation learning to develop a cost function that enhances an MPC controller. This system is designed to handle various social tasks, such as navigating around corners and maintaining appropriate interpersonal distances in a human-aware manner. However, unlike our approach, Performer-MPC requires training a separate cost function for each navigation scenario, limiting its generalizability. An adaptive cost learning strategy based on DRL has been proposed in [29]. Preliminary attempts to obtain a general planner for socially-aware navigation have been made in [8], however, treating the different scenarios with separate control solutions and introducing a classification step in the pipeline affects the smoothness of the overall system. Our study builds on and extends the work presented in [7]. This preliminary study focused on learning a cost for people groups and queues only, whereas this research aims at generalizing to a broader set of human-oriented scenarios, including people crossing, overtaking, corridors traversal, blind corners and evolving scenarios with occluded entrances. Moreover, the overall implementation has been revisited, and a general social cost layer approach has been developed to be easily integrated into the popular community framework Navigation 2 [30] as a standard plugin.

3 Background

Grid-based path planners, such as Dijkstra's and A*, operate on a discrete representation of the environment known as a grid map, which can be formalized as a two-dimensional, undirected graph $M = (V, E)$. In this graph, V represents the set of vertices with size $n \times m$, where each vertex corresponds to a specific cell within the map. The edge set E, of size k, consists of connections between adjacent cells. A path through this graph is defined as a sequence of vertices $P = (v_1, v_2, \ldots, v_n)$, where for each consecutive pair of vertices v_i and v_{i+1}, there exists an edge $e_{i,i+1} \in E$ with an associated distance $d_{i,i+1}$. This approach ensures a structured, efficient search for optimal paths while traversing the environment.

Given a starting vertex $v_s \in V$ and a goal vertex $v_g \in V$, the objective is to determine the shortest feasible path $P^* = (v_1, v_2, \ldots, v_n)$, where $v_1 = v_s$ and $v_n = v_g$. This optimal path minimizes the cumulative traversal cost defined by:

$$P^* = arg\, min_{v1,\ldots,vn \in V} \sum_{i=1}^{n-1} f_c(v_i, v_{i+1}), \tag{1}$$

where $f_c : V \times V \to \mathbb{R}^+$ is a *cost function* that assigns a non-negative value representing the traversal cost between consecutive vertices.

Cost functions can be constructed using multiple components that account for both the distance between nodes and the desirability or safety of traversing each node. A *cost map* C associated with a grid map M is represented as an $n \times m$ matrix, where each element $v_i \in V$ is assigned a cost $c_i \in \mathbb{R}^+$, indicating the cost of entering or occupying that node. Typically, prior knowledge of the environment, such as walls or static obstacles, is stored in a global cost map C_G. On the other hand, real-time sensor data capturing dynamic obstacles around the robot are integrated into a local cost

map C_L, which is centred on the robot's reference frame, and usually exploited by a local planner or controller. The total global planner cost of transitioning between nodes is defined as:

$$f_c(v_i, v_{i+1}) = d_{i,i+1} \cdot c_{G,i+1}, \tag{2}$$

where $c_{G,i+1}$ is the cost associated with v_{i+1} in the global cost maps.

4 Methodology

Our goal is to maintain the beneficial aspects of grid-based navigation systems, such as optimal pathfinding and obstacle avoidance, while incorporating socially compliant behaviours. To accomplish this, we introduce an additional cost map, named general social cost layer (GSCL), that specifically represents social factors, enabling the robot to avoid certain nodes in the environment based on social context. The social cost function $f_S : M_S \rightarrow C_S$ is learned through a deep neural network able to provide the *social cost map* C_S representing socially relevant costs. The neural network generates this output starting from the *social grid map* M_S that encodes all pertinent social information about the environment.

In this section, we first describe how we created the dataset used to train, test, and validate our network. Then, we present the architecture and training of our network. Finally, we describe the functionality of our social cost layer, explaining how it operates and what it predicts for each of the considered scenarios.

4.1 Costmap Dataset

We generate the dataset as pairs $(M_S^{(i)}, C_S^{(i)})$, where each pair consists of an input social grid map and its corresponding output social cost map. Both the input and output images are greyscale with a resolution of 40×40 pixels, where each pixel corresponds to an area of 40 cm by 40 cm. Namely, each image represents a 16×16 meter area centred around the robot.

The dataset is generated programmatically using an automatic generator we developed for this work. For each scenario, the algorithm produces an input image, namely the map M_S, by placing walls and humans according to the corresponding social norm. In these images, each element is encoded using specific pixel values. Walls are represented by pixels with a value of 255, while empty spaces are indicated by pixels with a value of 0. People are depicted through a combination of two adjacent pixels: a brighter pixel with a value of 100 to indicate the person's position and a lighter neighbouring pixel with a value of 25 to indicate their direction. The navigation goal is marked by a pixel with a value of 200. A certain degree of randomness is introduced for the generation of each scenario. This ensures variability for each input-output image pair, providing a richer and more representative dataset. The specific parameters governing this randomness for each scenario are detailed later in this section.

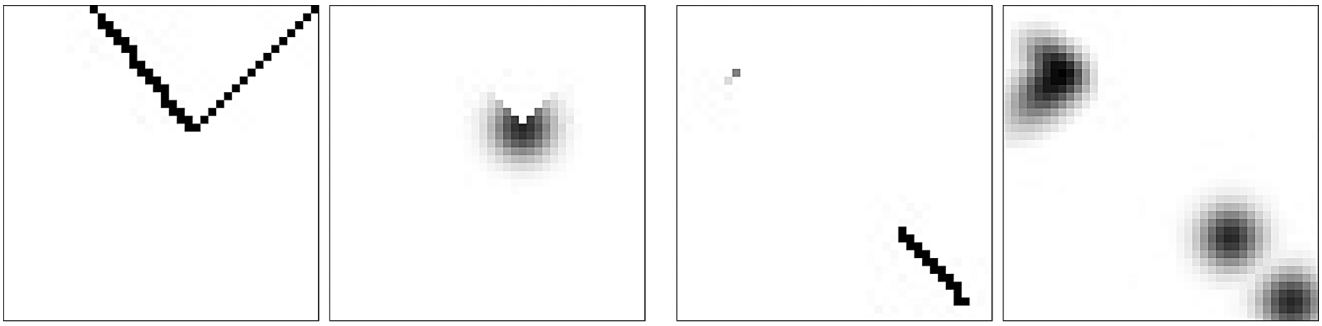
At the same time, the generator creates the corresponding output image, namely the cost map C_s, representing the social cost layer, with costs ranging from 0 to 255. Higher costs are assigned to areas surrounding individuals through a combination of Gaussian functions. Elevated costs are also applied to regions between groups of people, along the left side of hallways, and around corners. When the navigation goal is positioned in front of a line of people, increased costs are placed around the queue to reflect the social constraints. We apply our method to a series of social scenarios described below.

4.1.1 Corner

To ensure human safety, the robot should maintain a minimum distance of one meter from walls when approaching corners. This precaution helps prevent potential collisions if a human unexpectedly emerges from around the corner. As shown in Fig. 1(a), the input image consists solely of walls forming a corner. The social cost layer assigns a high cost around corners, encouraging the robot to avoid them and follow a wider trajectory. Randomization is introduced in the dataset by generating converging walls with random orientations and by randomly selecting the corner point within the image. The high cost is then applied around the obtuse angle. A total of 2000 samples are generated for this scenario.

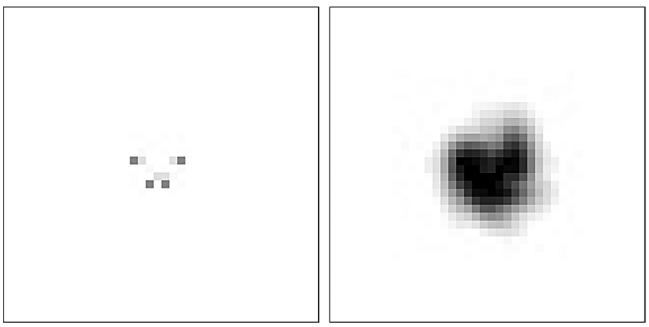
4.1.2 Proxemics Cost Around People

In this scenario, the input image includes the human pose and randomly generated walls. To respect personal space, the social cost output incorporates a pair of Gaussian functions around the human's position. This approach is similar to the ROS Social Navigation Layers (SNL) package [31], but with a key difference: in our implementation, the proxemics layer and the passing layer are combined into a single layer, as shown in Fig. 1(b). A total of 1000 samples are generated for this scenario.

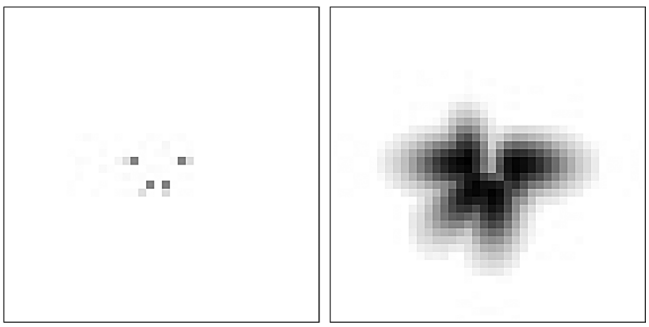


(a) *Corner scenario*

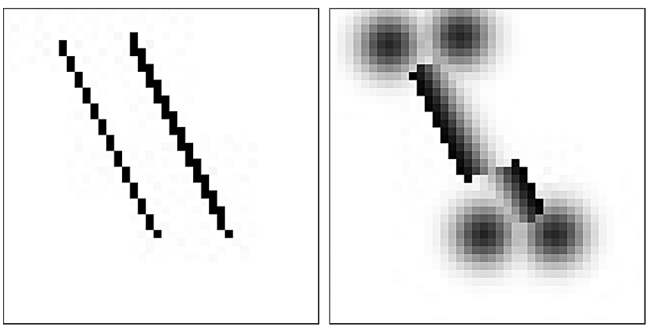
(b) *Single human scenario*



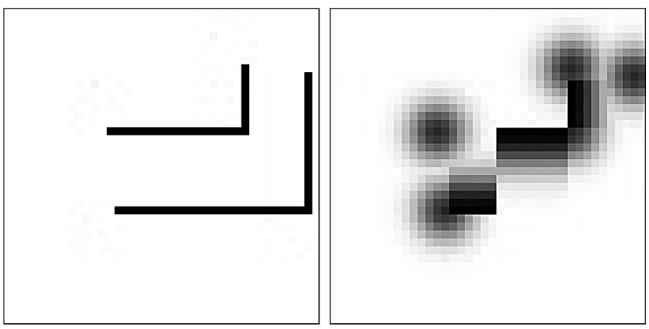
(c) *Group of people scenario*



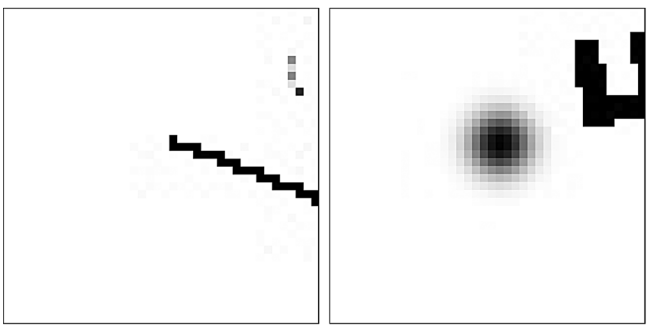
(d) *non-group of people scenario*



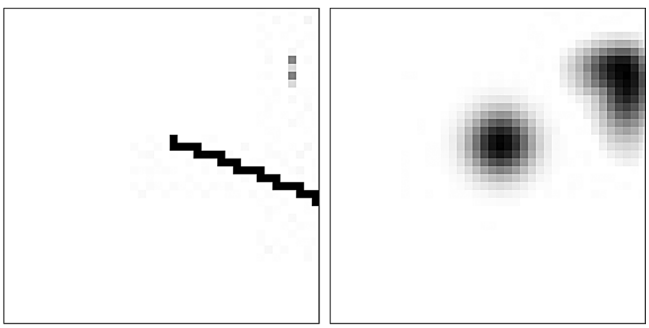
(e) *Hallway scenario*



(f) *L-shaped hallway scenario*



(g) *Queue of people scenario*



(h) *non-queue of people scenario*

Fig. 1 Input and target samples of the social cost layer in the generated training datasets. Different desired cost regions are defined according to social and safety rules for (**a**) blind corners, (**b**) stationary or walking humans, (**c-d**) groups and non-groups of people, (**e-f**) straight and L-shaped hallways, and (**g-h**) queue and non-queue scenarios

4.1.3 Wide Groups of People

When people form groups, the robot is expected to navigate around them rather than pass through the group. Kendon [32] and Marshall et al. [33] describe three types of F-formation for groups of two persons (L-shaped, face-to-face, and side-by-side) and three types for groups of more than two person (circular, semicircular, and triangular). In our study, within the input images of the group scenario, people are arranged in a wide group configuration, oriented toward one another. Each sample includes between 2 and 6 individuals, randomly distributed along a circumference with a radius ranging from 1 to 1.5 meters. This sampling strategy allows for the emergence of diverse group arrangements, effectively covering all six F-formations described in the literature. The corresponding social cost output includes human proxemics as well as an additional cost area within the group's interaction zone, as depicted in Fig. 1(c). To differentiate between groups and individuals, a dual-case scenario is generated. Figure 1(d) illustrates this case, where people are now positioned in a circle but oriented outward. In this scenario, only Gaussian proxemics costs are applied, reflecting the lack of an interaction area within the group. We generated 4500 samples for the group scenario and 4500 for the non-group scenario.

4.1.4 Hallway

When navigating through a hallway, the robot is expected to keep the right side. To achieve this behaviour, we generate a higher social cost on the left side of the hallway, regardless of the robot's direction. This approach is applied to both straight and L-shaped hallways, as illustrated in Fig. 1(e) and (f), respectively. The increased cost around the corner is also clearly visible in the case of L-shaped hallways. Hallway walls are generated with random lengths, orientations, and widths, ensuring the hallway's centre line aligns with the centre of the image. In addition, we consider a third case representing straight hallways with people passing by. For this scenario, we generated a total of 5000 samples, distributed across the three hallway configurations.

4.1.5 Queue

In the queuing scenario, the robot must queue behind the last person in line to reach the goal, ensuring it does not cut ahead. The input image depicts a line of people directed towards the navigation goal. Figure 1(g) shows this case, where the target social cost includes a U-shaped area around the queue that prevents the robot from cutting in line and instead guides it to the end of the queue. We generate a dual case to differentiate between situations where the robot should queue and when it should not. As shown in Fig. 1(h), if the navigation goal is not in front of the queue but elsewhere, the social cost layer will only include proxemics costs around people since there is no need for the robot to queue. Queues are generated with a random orientation and a length ranging from 3 to 10 meters. For each input-target sample, the average distance between individuals is drawn from a triangular distribution with limits of 0.3 and 0.8 meters. Walls are then randomly positioned, ensuring they do not intersect the queue. We generated 3600 samples for the queue scenario and an additional 3600 for the non-queue scenario.

In these scenarios, we focus on encoding proxemics and other social conventions that depend solely on the spatial configuration of people and the environment, rather than on the robot's direction of approach. For this reason, modelling costs as cell-dependent (independent of the incoming direction) is sufficient to capture the intended social behaviours.

It is important to clarify that the generator can only produce predefined pairs $(M_S^{(i)}, C_S^{(i)})$ for each scenario. However, this is not equivalent to learning a general function f_S that maps a general social grid map into its corresponding cost map. Hence, the use of an automatic algorithmic generator to create the dataset does not prevent the need to learn a neural model of the function f_S. Moreover, the generator can only compute the social cost $C_S^{(i)}$ knowing what scenario $M_S^{(i)}$ is considered in advance; it cannot recognize the scenario from raw measurements of people and walls. Besides, the algorithm follows a fixed, deterministic pattern in generating both the scenario M_S (i.e., the spatial arrangement of people, walls, and goals) and the associated social cost map C_S simultaneously, by directly encoding prior knowledge of the ground-truth social context (e.g., whether the people are in a queue, forming a group, etc.). As a consequence, it cannot generalize to variations of the same scenario that deviate from this strict pattern. In contrast, the neural network learns the function f_S, enabling it not only to recognize a given scenario, or multiple scenarios when they co-occur, but also to generalize the generation of C_S to similar unseen scenarios. Another key difference lies in flexibility: while the generator requires a dedicated script for each individual social norm to be encoded, a single neural network can capture and integrate multiple social norms within the same model. For this reason, the algorithmic generator is used exclusively to produce training samples for learning f_S, while the neural network provides the capacity to infer from any given grid map the appropriate cost map across different scenarios with a unique model.

4.2 Neural Network Architecture

The encoder-decoder U-Net architecture used in this work consists of convolutional layers for feature extraction, max-pooling layers for down-sampling, and transpose convolutional layers to reconstruct the output to its original resolution. The network contains a total of 1.616.642 learnable parameters, and its structure is depicted in Fig. 2.

In the encoder, the 40×40 pixel gray-scale input images is progressively down-sampled to capture multi-scale features. This process involves three convolutional blocks, each containing two convolutional layers with a kernel size of 3 and padding of 1. Each convolution block is followed by batch normalization and ReLU activation. Max-pooling layers reduce the spatial dimensions, allowing the model to extract high-level representations while preserving spatial coherence. A final convolutional block serves as a bottleneck, maintaining the network's capacity to learn complex features.

In the decoder, transposed convolutional layers with a kernel size of 2 and a stride of 2 are used to up-sample feature maps and to restore spatial resolution. Skip connections are employed to integrate information from both the deep and shallow layers, preserving fine-grained details, improving gradient flow, and mitigating information loss during down-sampling. Batch normalization and ReLU activation are applied after each up-sampling step. At the end, the output image obtained is again a 40×40 pixel grayscale image, representing the social cost map.

The dataset is split into training and validation sets using a 70%-30% ratio. Training is conducted over 20 epochs using the ADAM optimizer with a learning rate of 0.001. The training process is carried out on a machine with an Intel Core i7-9700K CPU and an Nvidia GeForce RTX 2080 Ti, taking approximately 5 minutes. In our experiments, we evaluated the network using pixel-wise mean squared error (MSE) over normalized cost map results. The results showed an average training MSE of 0.0021 and a test MSE of 0.0033.

4.3 General Social Cost Layer

Figure 3 illustrates the proposed methodology. The people's positions and orientation, and the goal are incorporated into a social local grid map M_S, which is centred on the robot's reference frame. The problem is posed as an image-to-image translation problem, where the trained encoder-decoder model processes this map to produce the social cost map C_S. This map is then combined with other navigation cost layers to generate the overall cost map used for path planning. This process continuously updates as the social cost map is refreshed to reflect the evolving positions of individuals within the scene, ensuring dynamic adaptation to changing social contexts.

The total cost of transition between two nodes (v_i, v_{i+1}) defined in Eq. 2 can be extended to incorporate social factors, resulting in the following formulation:

$$f_c(v_i, v_{i+1}) = d_{i,i+1} \cdot \max(c_{G,i+1}, c_{S,i+1}), \tag{3}$$

where $c_{S,i+1}$ represents the additional cost derived from the estimated social cost map C_S, and max is the cell-wise maximum function used to combine the different cost layers and preserve safety keeping the highest cost for each map cell.

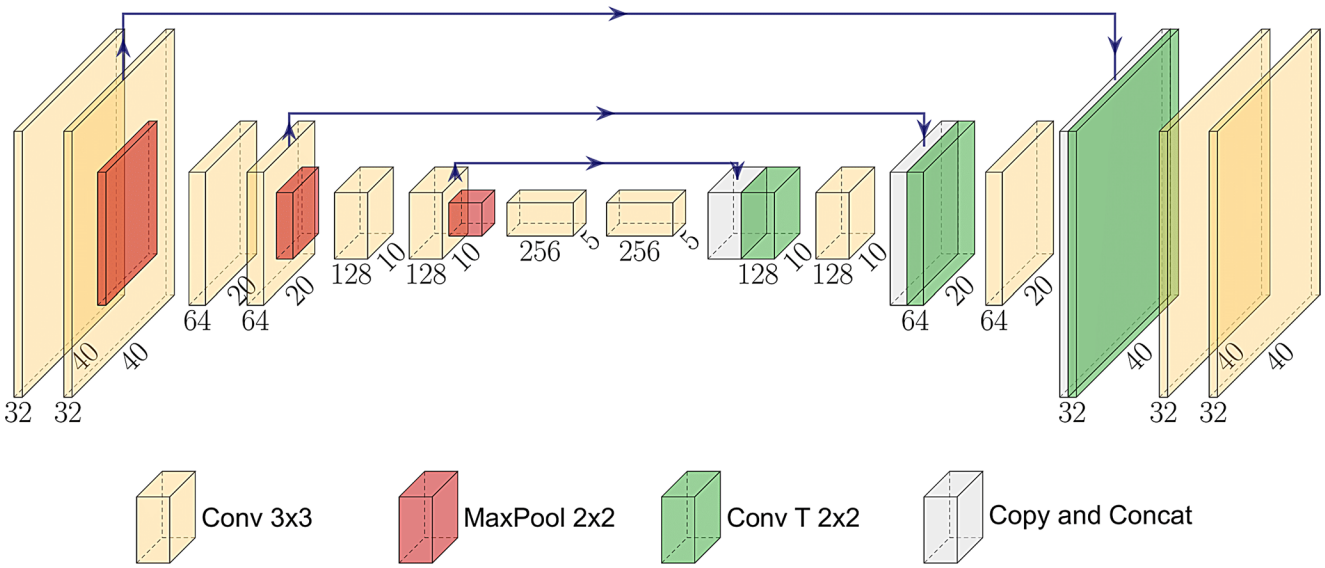


Fig. 2 Neural network architecture: an encoder-decoder convolutional structure with residual connections is adopted to predict the social cost layer from an input map containing people

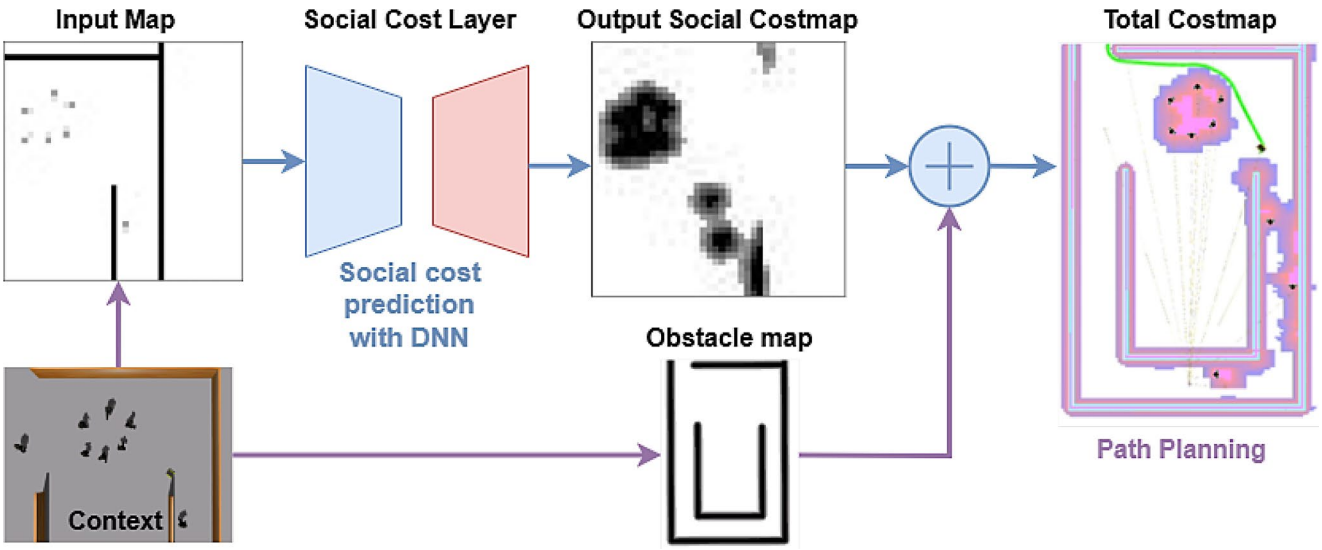


Fig. 3 Schematic of the proposed general social cost layer approach: context information about people in an area of 16×16 meters is encoded in an input map. An encoder-decoder DNN is trained to predict a social costmap layer to be added to the classic obstacle layer used for navigation. A classic grid-based planner is finally run to compute socially-aware path

To generate the input image, we first convert the entire occupancy grid map into a 40 cm/pixel image, setting pixel values above 0.6 to 255 and those below to 0. Next, we add people to the image: their positions are marked as pixels with a value of 100, while a secondary pixel with a value of 25 indicates their orientation. This input image is then fed to the U-Net for inference.

The U-Net output is then processed to create the social cost layer. It is first rescaled to match the cost map's resolution and then placed onto an empty cost layer. The resulting layer, which is zero everywhere except in the area surrounding the robot, is provided to the planner to be integrated into the navigation process.

4.4 Deployment

We implemented the proposed method both in simulation and on a real robot by integrating it into a ROS 2 [34] node. This node periodically gathers the positions of the navigation goal and surrounding people, converting this information into the input grid map image representation M_S required by the network. The generated grid map image is then processed by the neural network, which outputs the predicted social cost layer C_S. The resulting layer is rescaled and published as an OccupancyGrid on a ROS 2 topic, making it accessible to the navigation system using a Nav 2 cost map plugin. The Nav 2 [30] planner Nav-Fn algorithm will then perform its task using a global cost map, which includes the social cost layer C_S along with the other layers provided by the navigation framework, in particular, the static and the inflation layer. When combining cost layers, Nav 2 employs a `updateWithMax` method, which applies a cell-wise maximum across all active layers. As a result, obstacle-related costs always dominate when present, thereby ensuring that collision avoidance consistently takes precedence over social preferences.

The adopted Nav-Fn planner consists of a wavefront Dijkstra expanded holonomic planner coupled with a Model Predictive Path Integral Controller (MPPI) to plan and execute socially-aware navigation in dynamic environments effectively. The overall navigation task is managed by a behaviour tree, which handles the re-planning at $1Hz$. In the special case of the queue scenario, a simple stop-and-go mechanism based on the distance to the next person in line is used to ensure that the robot proceeds to the goal in a socially appropriate manner.

5 Experiments and Results

Our solution has been extensively validated with a series of navigation experiments in simulation environments and in real-world conditions. The experimental scenarios were designed using a combination of isolated and hybrid social situations, as defined in Sect. 4.1. Throughout this section, we compare our proposed General Social Cost Layer (GSCL) against two baselines: the standard Nav-Fn planner without any social layer, and the ROS 2 Social Navigation Layers (SNL) [31] with two different configurations. The two configurations differ in the covariance parameter for the Gaussian functions around people: smaller in the first case (SNLσ_1) and larger in the second (SNLσ_2). This setup was chosen to illustrate a key limitation of the SNL, namely the need for scenario-specific parameter tuning. While adjusting

the covariance can yield better performance in some cases, it often degrades performance in others. By contrast, our GSCL achieves consistently strong results across all scenarios without any parameter tuning, demonstrating both its robustness and generalizability.

In Sect. 5.1, we define the metrics used to evaluate the robot's navigation from a social perspective. The experimental setup and results for the simulation environment are presented in Sects. 5.2 and 5.3, while Sects. 5.4 and 5.5 detail the real-world experiments and their outcomes.

5.1 Evaluation Metrics

In global path planning, the primary concern is the overall route of the robot rather than local, moment-to-moment commanded action. In the scope of this research, the key challenge lies in ensuring that the planned path adheres to social conventions, such as avoiding the disruption of groups, following right-side navigation in hallways, and respecting queueing behaviours. To evaluate these aspects, we adopted quantitative metrics to capture the efficiency of navigation and the adherence to social rules. These metrics, listed below, are adopted from Mavrogiannis et al. [4], and represent the standard for evaluations of social-compliant path planning.

5.1.1 Goal-oriented Success Rate

Quantifies the percentage of trials in which the robot successfully reached the goal. This is the opposite of the "Failure rate" as presented in Mavrogiannis et al.

5.1.2 Social-oriented Success Rate

Referred to as the "Social score" in Mavrogiannis et al., this metric quantifies the percentage of trials in which the robot successfully reached the goal while adhering to the expected social behaviours.

5.1.3 Proxemics

Quantifies the percentage of time the robot spent within each of the four proxemic zones: intimate, personal, social and public spaces. In our study, we adopted the proxemics model introduced by Hall E. T. [35], which remains the most widely accepted standard in social robot navigation [9]. It is worth clarifying that proxemics measurements are strongly influenced by the controller's behaviour rather than by the path planner itself. Consequently, the goal-oriented and social-oriented success rates should be regarded as the primary quantitative metrics of this study, while proxemics are reported for completeness and additional insight.

However, since social norms are difficult to quantify mathematically, algorithmic metrics may not fully capture the effectiveness of socially aware navigation. To address this, this study also employs a human-centred survey approach, leveraging human evaluators to assess the social acceptability of the robot's paths. The metrics chosen for the survey evaluation are described below. Also in this case, these metrics are adopted from [4].

5.1.4 Social Compliance

How well the robot's navigation adheres to expected social norms, maintains appropriate distances, does not cut through groups, holds the right in corridors, and correctly joins the line.

5.1.5 Naturalness

Referred to as "behavioural Naturalness" in Mavrogiannis et al., evaluates how much the robot's path appears natural, smooth, and comfortable, or if it seems jerky, unnatural, or forced.

5.1.6 Safety

Evaluates how safe the robot's navigation results are, and how well potential conflicts with humans are avoided in advance.

5.1.7 Predictability

Referred to as "Legibility" in Mavrogiannis et al., this metric quantifies how predictable the robot's path is, or whether it takes unexpected or confusing directions.

It is worth mentioning that the lack and limitation of well-established metrics for socially compliant path planning is a well-recognized challenge within the robotics community. However, the metrics included in our study are among the few capable of effectively assessing socially aware navigation, as discussed in Mavrogiannis et al. [4].

5.2 Validation in Simulation

The performance of the social planning system has been validated in a realistic simulation first. Different environments and scenarios have been considered, consisting of a mixed series of hallways, corners, wide open spaces, and narrow passages, simulating a typical indoor setting such as an office or a shopping mall. In this context, some human agents are walking through hallways, passages, and open spaces, while others are standing in groups or in queues. The Gazebo simulator, coupled with the HuNavSim [36]

plugin, has been used as simulation framework. The validation scenarios are briefly described below:

5.2.1 L-shaped Hallway

In this scenario, the robot navigates an L-shaped hallway (Fig. 4(a)). The planner is expected to keep the right side of the hallway while avoiding a person located just around the corner. Since the person is initially hidden from view, the robot cannot detect them in advance. The expected behaviour, both for a human and for the robot, is to anticipate the potential presence of a person behind the corner and approach cautiously, following a wider, more conservative curve.

5.2.2 Group in Open Space

In this scenario, the robot traverses an hallway, then it crosses an open space with an interacting group (Fig. 4(g)). It is designed to showcase how the planner deals with people moving and standing in hallways, even not necessarily respecting the keep right convention, and with people forming a group. We expect the social planner to keep right in the hallway, except for when people are present on the same side without collaborating, and to pass around the group.

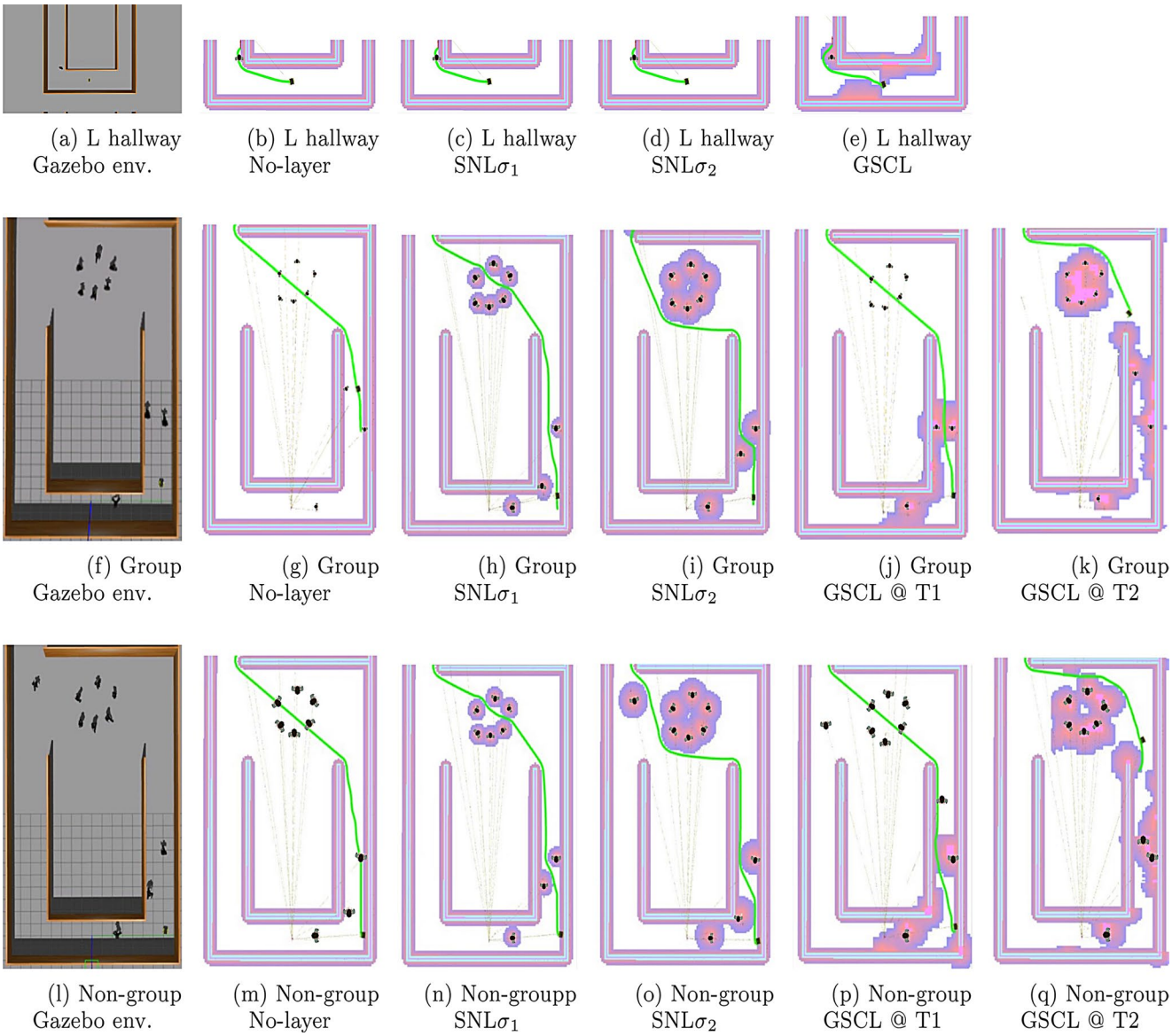


Fig. 4 Gazebo simulation environments and Rviz visualization of the costmap and planned path obtained in L-hallway, group and non-group scenarios. The global costmap and the path computed with the naive Nav-Fn planner with the ROS SNL using two different covariance values (third and fourth columns), and with the GSCL (fifth and sixth columns), showing two key time instant of replanning

5.2.3 Non-group

This scenario is similar to the previous one, but people arranged in a circle are oriented outward (Fig. 4(m)). Since they are not interacting, it is socially acceptable to either pass through the group or navigate around it. However, proxemics boundaries of each individual should always be respected.

5.2.4 Crowded Entrance

The robot navigates from a crowded open space to an enclosed area accessible via two entrances (Fig. 5(a)). The left entrance, which is initially free and allows for a shorter path, becomes too congested at a certain time due to the presence of two walking persons. The social planner is expected to go through the less crowded right entrance, although increasing the path length, without heading towards humans and forcing them to move away to let it pass.

5.2.5 Queue

The robot starts in an open space with people queuing in front of a target desk, and it must reach its front (Fig. 5(g)). The social planner is expected to respect the queuing convention, hence planning to the goal through the end of the queue as a person would do.

5.3 Simulation Results

For each scenario and each algorithm, we conducted 30 independent trials. The main quantitative results of the simulation experiments are summarized in Table 1. This table gather the success rate of the navigation in terms of goal-oriented task and expected social behaviour, with percentages referring to the total number of trials. Complementarily, Fig. 6 presents the proxemics results, obtained by averaging outcomes across all trials. Table 2 reports the results of Games–Howell post hoc pairwise comparisons, performed after the Welch's ANOVA conducted on the proxemics measurements. This analysis was selected to identify statistically significant differences between the proposed GSCL group and the other groups. This test was chosen since it accounts for heteroscedasticity highlighted by the Levene Test and the unequal sample sizes.

The qualitative performance of the simulation scenarios has been assessed by submitting the questionnaire to the participants and collecting their evaluation scores, with the metrics defined in Sect. 5.1. For each scenario and each algorithm, the survey presented a series of images of the considered situations in the simulation environment, together with

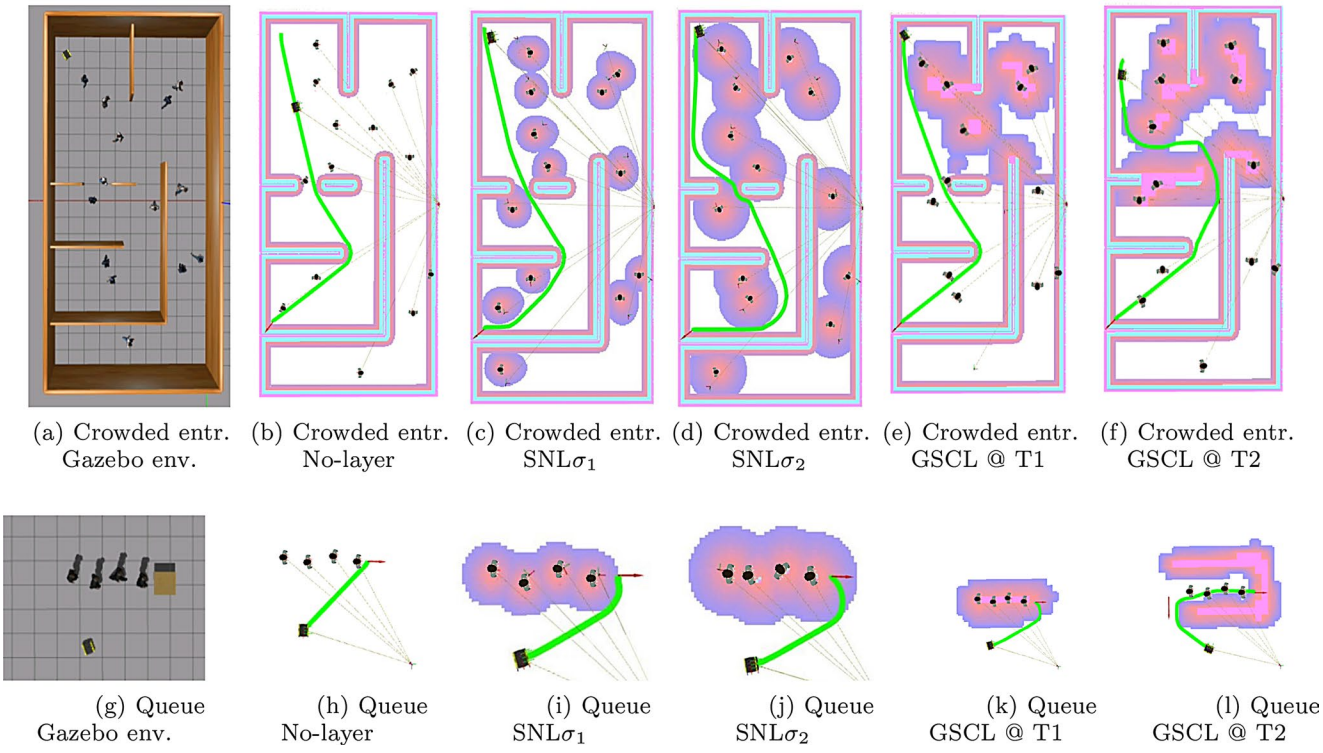


Fig. 5 Gazebo simulation environments and Rviz visualization of the costmap and planned path obtained in Crowded entrance and queue scenarios. The global costmap and the path computed with the naive Nav-Fn planner with the ROS SNL using two different covariance values (third and fourth columns), and with the GSCL (fifth and sixth columns), showing two key time instant of replanning

Table 1 Success rate results from the simulation experiments, reported in terms of goal-oriented task completion and adherence to expected social behaviors

Env	Method	Goal [%]	Social [%]
L Hallway	No − Layer	43	16
	SNLσ_1	40	23
	SNLσ_2	40	20
	GSCL	100	100
Group	No − Layer	70	0
	SNLσ_1	100	0
	SNLσ_2	100	100
	GSCL	100	100
Non Group	No − Layer	70	-
	SNLσ_1	100	-
	SNLσ_2	100	-
	GSCL	100	-
Crowded	No − Layer	93	0
Entrance	SNLσ_1	93	60
	SNLσ_2	97	36
	GSCL	100	93
Queue	No − Layer	100	0
	SNLσ_1	100	0
	SNLσ_2	100	0
	GSCL	100	93

The comparison covers the five considered scenarios and four navigation strategies: the baseline Nav-Fn planner, the ROS 2 SNL with low and high covariance settings, and the proposed GSCL

the corresponding cost maps and planned paths (similar to the visualization shown in Figs. 4 and 5), a short description of the scenario, and an explanation of each evaluation metric. For each metric, participants rated the robot's behaviour on a five-point (1–5) Likert scale. In addition to isolated scenarios, the survey also included a set of mixed scenarios, where multiple social norms appear. The survey involved 50 participants from Italy, Spain, and the United Kingdom, ranging in age from 20 to 65 years, with a mean age of 41. Approximately 48% of the participants were female and 52% male. Nearly 64% rated themselves as having low background knowledge in robotics and autonomous navigation (up to 2 out of 5), while about 24% reported high expertise (4 or 5 out of 5). The survey followed a within-subjects design: all participants received the same questionnaire and responded to the same set of questions. All results were obtained by calculating the mean and standard deviation for each metric across the total collection of answers. Figure 7 presents these results as histograms, along with pairwise comparisons from Tukey's HSD test, with statistical significance denoted by asterisks corresponding to p-value thresholds following the GraphPad convention [37].

5.3.1 L-shaped Hallway

As shown in Fig. 4(b–d), the planned path is identical for the baseline Nav-Fn planner and for both the small- and large-covariance SNL configurations. This occurs because the person remains hidden behind the corner for the entire corridor traversal, preventing the ROS 2 SNL from perceiving them or anticipating their presence. In all three cases, the planned path leads directly into a collision. Avoidance is therefore left entirely to the controller, which will realistically responds by braking abruptly at the corner. Such late reactions may startle the person and result in socially unacceptable behavior.

In contrast, the GSCL (Figs. 4(e) and 8(a)) guides the robot to keep the right side of the corridor and assigns a high cost around the corner. This encourages the robot to follow a wider trajectory, proactively accounting for the possible presence of a person.

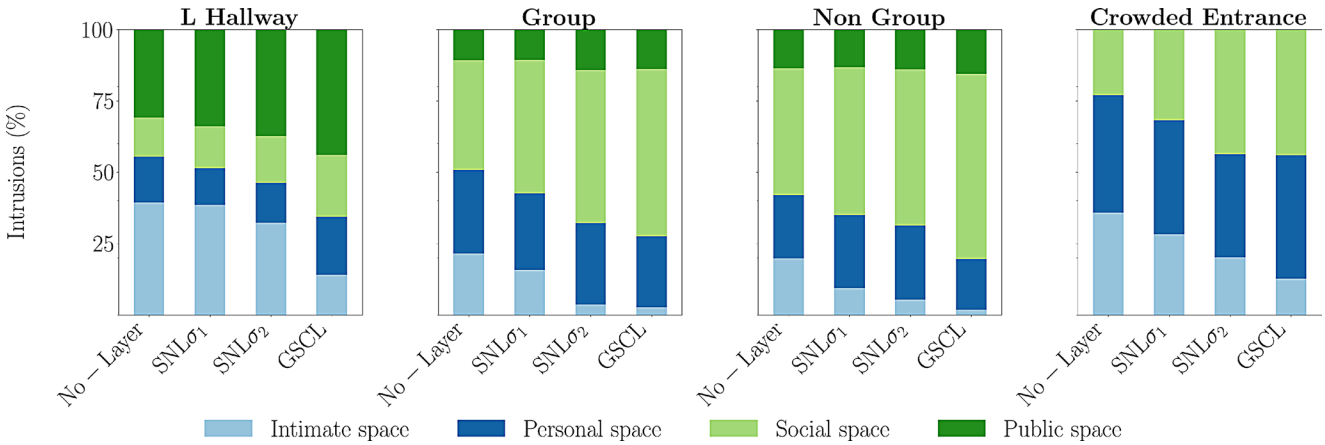


Fig. 6 Proxemics results from the simulation experiments across the four considered scenarios. Proxemics is measured as the percentage of time the robot remains within the four spatial zones around people. The comparison includes navigation using the baseline Nav-Fn planner, the ROS 2 SNL configured with both low and high covariance values, and the proposed GSCL

Table 2 Results of Games–Howell post hoc pairwise comparisons following Welch's ANOVA

Env	Intrusion	Method	p-value	Hedges g	Effect Sizes
L Hallway	Intimate	No − Layer	**0.0050**	−2.058	very large
		SNLσ_1	**0.0018**	−2.497	very large
		SNLσ_2	**0.0043**	−2.161	very large
	Intimate + Personal	No − Layer	**0.0022**	−2.333	very large
		SNLσ_1	**0.0046**	−2.305	very large
		SNLσ_2	**0.0189**	−1.837	very large
Group	Intimate	No − Layer	$\leq$ **.0001**	−8.774	very large
		SNLσ_1	$\leq$ **.0001**	−6.641	very large
		SNLσ_2	0.4680	−0.372	small
	Intimate + Personal	No − Layer	$\leq$ **.0001**	−4.508	very large
		SNLσ_1	$\leq$ **.0001**	−3.535	very large
		SNLσ_2	**0.0010**	−1.052	large
Non-Group	Intimate	No − Layer	**0.0362**	−10.770	very large
		SNLσ_1	$\leq$ **.0001**	−2.220	very large
		SNLσ_2	$\leq$ **.0001**	−1.306	very large
	Intimate + Personal	No − Layer	$\leq$ **.0001**	−5.926	very large
		SNLσ_1	$\leq$ **.0001**	−4.781	very large
		SNLσ_2	$\leq$ **.0001**	−2.975	very large
Crowded	Intimate	No − Layer	$\leq$ **.0001**	−3.195	very large
Entrance		SNLσ_1	$\leq$ **.0001**	−2.213	very large
		SNLσ_2	**0.0012**	−0.969	large
	Intimate + Personal	No − Layer	$\leq$ **.0001**	−2.808	very large
		SNLσ_1	$\leq$ **.0001**	−1.340	very large
		SNLσ_2	0.9977	−0.047	negligible

Each row compares the listed method with our GSCL. The table reports the adjusted *p*-value and Hedges' *g* effect size; the sign of Hedges' g corresponds to the direction of the mean difference. All reported effect sizes are negative, indicating that GSCL reduces the time spent in the intimate or intimate and personal zones

This difference is clearly reflected in both the quantitative and survey results. From the proxemics analysis in Fig. 6 and Table 2, the GSCL spends significantly less time in both intimate and personal spaces compared to the other methods. Moreover, as reported in Table 1, the GSCL enable the robot to always reach its goal in a social compliant manner, while the other planners frequently result in collisions that prevent goal completion. Finally, the survey results in Fig. 7 further highlight the effectiveness of our GSCL solution, which substantially outperform all other approaches across all four metrics: social compliance, naturalness, safety, and predictability.

5.3.2 Group in Open Space

For the GSCL case, at the first time instant T1 (Figs. 4(j), and 8(b)), the system assigns higher costs to the left side of the hallway and around nearby humans. The robot initially follows the right-side convention, then deviates to overtake a person on the right, prioritizing personal space. When reaching the open space at time T2 (Figs. 4(k), and 8(c)), the robot correctly identifies the interacting group and adapts its trajectory to navigate around it in a socially compliant manner.

For the SNL case, changing the covariance value does not significantly affect hallway traversal (Fig. 4(h,i)). However, the robot does not keep to the right and instead occupies the centre of the hallway. Once the group is reached, instead, covariance plays a crucial role: with a larger covariance, the robot successfully detours around the group, whereas with a smaller covariance, the planner cuts through the group, violating the social norm. In the baseline case without any social cost layer (Fig. 4(g)), the robot cuts in front of people in the hallway and directly through the group.

These qualitative outcomes are consistent with the quantitative results in Table 1, Fig. 6, and Table 2, as well as the survey scores in Fig. 7. Overall, the GSCL achieves performance comparable to the SNL configured with a large covariance, with the former performing better on the proxemics scores. Both the baseline Nav-Fn planner and the small-covariance SNL configuration perform significantly worse across all metrics.

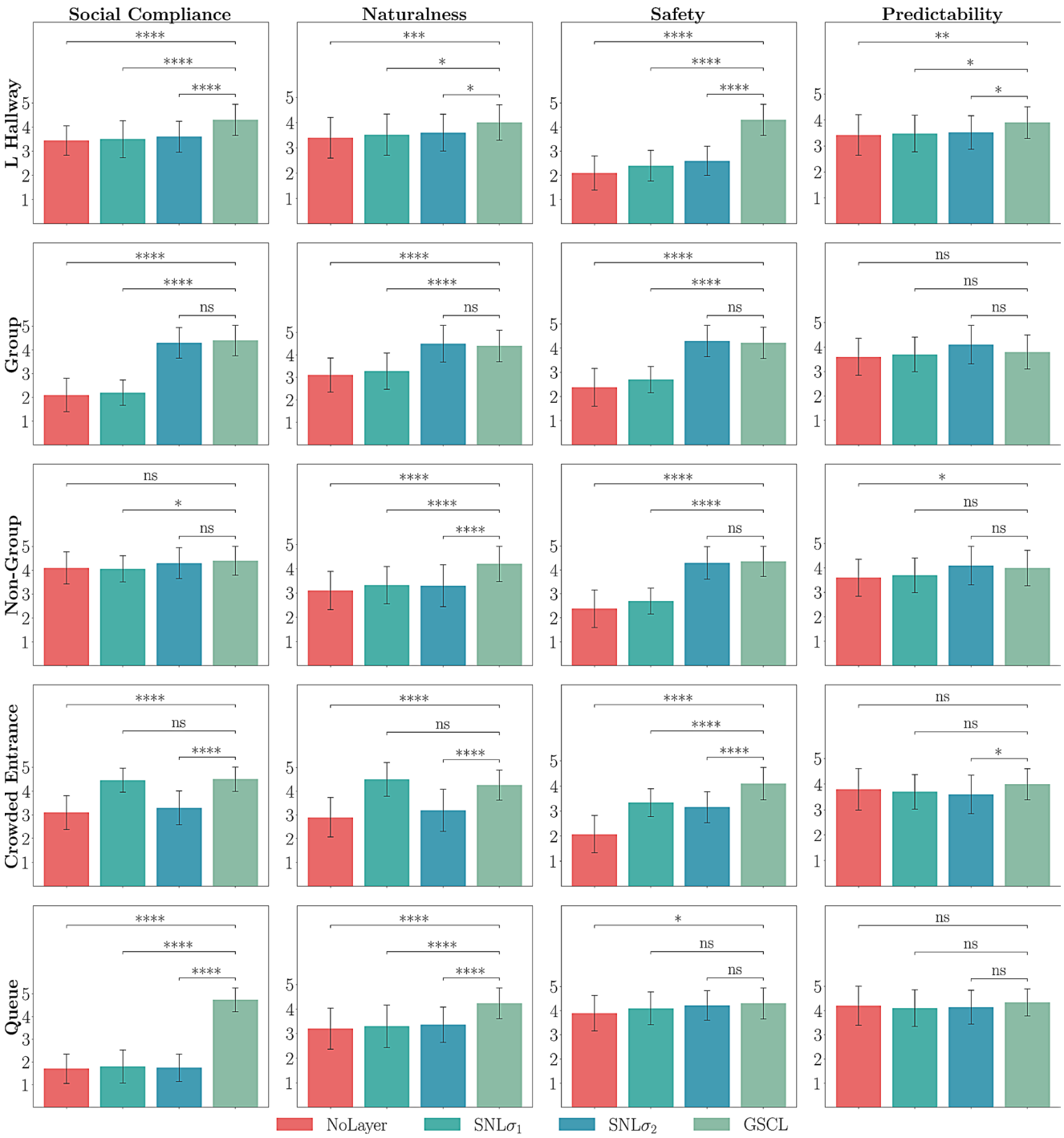
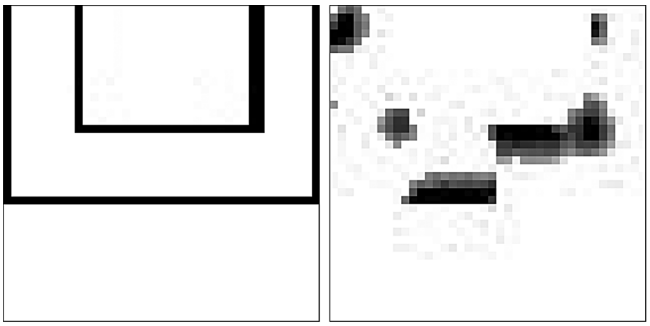
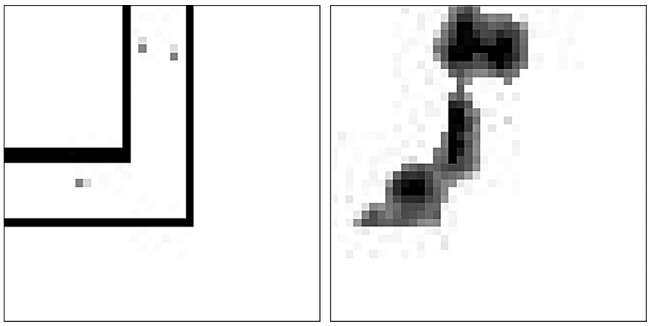


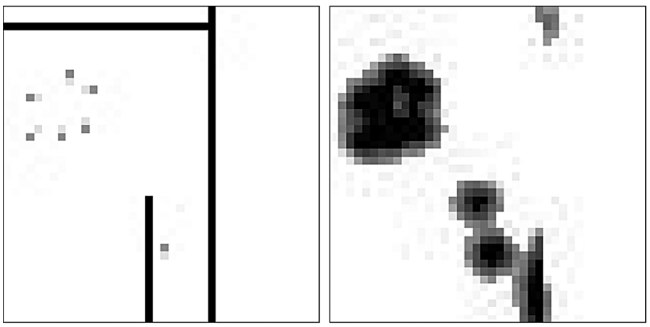
Fig. 7 Evaluation scores of simulation experiments collected from the survey. Turquoise and blue columns correspond to the ROS 2 SNL, with smaller and larger covariance values, respectively. Green columns indicate the results obtained with the proposed GSCL, while red columns represent the baseline Nav-Fn planner without any social layer. Reported mean values and standard deviations across multiple experimental runs highlight the overall improvement in the robot's social behaviour achieved with our approach. The asterisks (*) indicate the *p*-values obtained after performing the pairwise Tukey HSD test



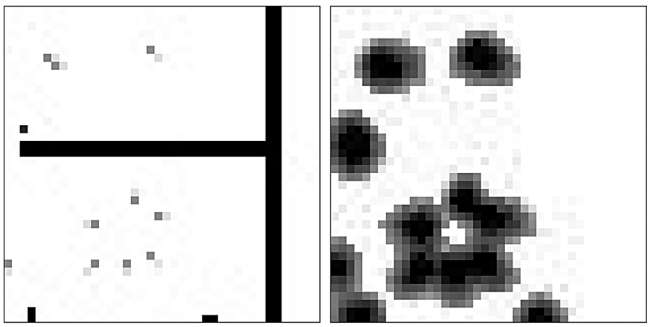
(a) *L hallway (corridor cost)*



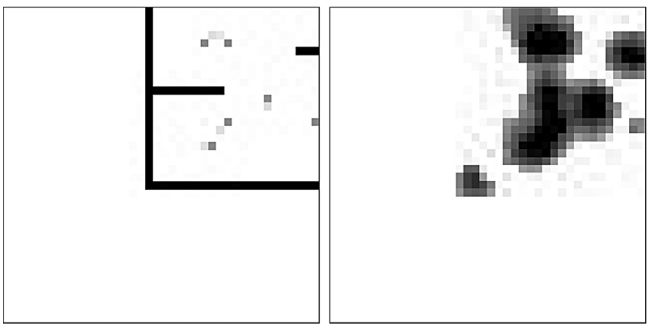
(b) *Group in open space @ T1 (corridor and people costs)*



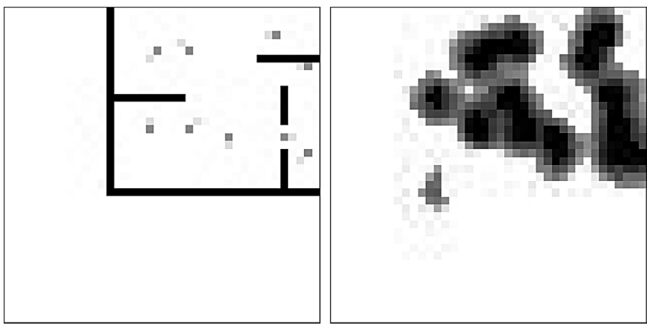
(c) *Group in open space @ T2 (group cost)*



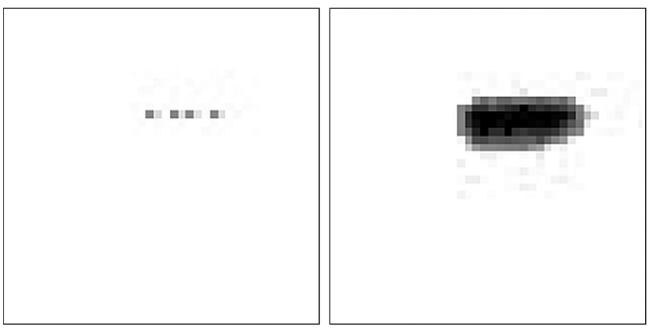
(d) *Non-group (proxemics costs)*



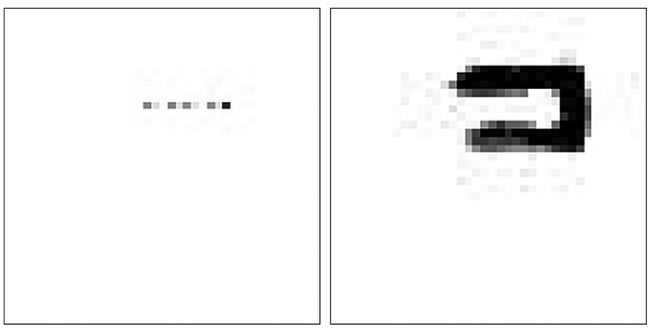
(e) *Crowded entrance @ T1 (crowded entrance cost)*



(f) *Crowded entrance @ T2 (crowded entrance cost)*



(g) *Queue @ T1 (non-queue cost)*



(h) *Queue @ T2 (queue cost)*

Fig. 8 Input and output maps of the social cost DNN at the time instants T1 and T2 showcased in Figs. 4 and 5. Brighter pixels in the output map indicate higher cost. The new social cost predicted at T2 allows the robot to replan and handle correctly the group, the crowded entrance and the queue, which is triggered by the goal assignment

5.3.3 Non-group

In this case, the trajectories planned by the baseline Nav-Fn with no social layers and both the SNL configurations are identical to the previous scenarios, as shown in Fig. 4(m-o). Since the position of people remains unchanged, the orientation alone does not affect the Nav-Fn behaviour. Moreover, since people are not moving, their orientation does not alter the behaviour of ROS 2 SNL either.

The only visible difference emerges with the GSCL case at time T2, where the robot reaches the group. As shown in Figs. 4(q) and 8(d), the cost layer enforces proxemics around people but not within the group. This means that the planned path bypasses the individuals to respect proxemics. However, if people were slightly farther apart, the planner might have routed the robot between people, since the network correctly recognized the absence of an interacting group configuration.

Accordingly, the goal success rate reported in Table 1 is identical to the previous case, while the social-based success rate is omitted, as no specific social norm applies in this scenario. Survey results indicate that all the algorithms present comparable results, though GSCL performs better in terms of safety and naturalness, as highlighted in Fig. 7. In addition, proxemics analysis shows that GSCL spends the least amount of time in intimate and personal spaces compared to the other methods (Fig. 6 and Table 2).

5.3.4 Crowded Entrance

In this scenario, the robot must choose between two possible passages to cross a crowded space. Both the SNL configurations (Fig. 5(c,d)) and the baseline Nav-Fn planner alone (Fig. 5(b)) only consider the left entrance in the map, since this is the shortest way. However, this passage is occupied by two people, causing the planned trajectory to cut in front of them.

In contrast, with the GSCL, the robot initially plans to enter through the left entrance (Fig. 5(e)), since the right entrance is beyond the social cost layer map extension (Fig. 8(e)). Then, both entrances become visible (Fig. 8(f)), and the neural network assigns a high cost to the crowded left entrance, leading the robot to choose the less congested right entrance (Fig. 5(f)). This results in a longer but more socially acceptable trajectory.

Again, these qualitative outcomes align with the quantitative and survey results. As well evident from Fig. 7, contrary to the group scenario, in this case the GSCL performance is comparable with the small-covariance SNL, while the large covariance-SNL performs worse. This result demonstrates that the GSCL consistently performs well without the need for scenario-specific parameter tuning, whereas the SNL requires careful adjustment of covariance values. Additionally, the GSCL outperforms all alternatives in terms of safety, goal-oriented success rate, and social-oriented success rate (Table 1), while also minimizing the time spent in intimate space (Fig. 6 and Table 2).

5.3.5 Queue

In this scenario, as shown in Fig. 5(h-j), the paths planned by the baseline Nav-Fn planner and both SNL configurations (small- and large-covariance) direct the robot directly toward the goal, cutting the line and violating the queueing norm. This behaviour is quite predictable since, without a dedicated algorithm, the simple Nav-Fn planner is not only unable to follow a queue, but also to recognize it altogether.

Conversely, GSCL behaves adaptively. As shown in Figs. 5(k) and 8(g), when the goal is not at the head of the queue, the network only enforces proxemics around individuals, without assigning high costs to the queue as a whole. In this way, if the goal is given elsewhere, the robot is free to bypass the queue at any point, including between people in line if possible and necessary, as a human would do. However, once the goal is set at the front of the line, the network recognizes the queue and generates the corresponding cost structure, visible in Figs. 5(l) and 8(h). In such a way, the robot is forced to enter the queue from the beginning and wait for its turn.

The quantitative and survey results confirm these observations. Table 1 shows that while all methods reach the goal, only GSCL respects the queueing norm. Likewise, the survey results in Fig. 7 demonstrate that GSCL outperforms the alternatives across social compliance and naturalness metrics.

In this case, the proxemics results are omitted as they are less meaningful. In fact, the queue scenario represents one of the few cases where the robot is actually expected to stay close to people in line, and the time spent in their personal space is more related to the velocity of advancement of the queue rather than the quality of the navigation.

5.4 Real-world Validation

Real-world experiments have been conducted in a laboratory equipped with a Vicon multi-camera tracking system to precisely record the poses and trajectories of both robot and walking people. The robot used for the test is a Jackal from ClearPath Robotics, limiting its maximum linear velocity to $v \in [0.0, 0.6] m/s$ and its maximum angular velocity to $\omega \in [-1.5, 1.5] rad/s$. The entire navigation system runs on the embedded computer of the robot, mounting an Intel i5 CPU as the main processing unit, and a 2D Lidar for obstacle detection.

Three maps were designed in the lab experiments: an L-shaped hallway, an open space with interacting groups, non-groups, and people queuing, and two rooms connected by two narrow entrances dynamically occluded by walking pedestrians. From these maps, we extracted the five scenarios previously introduced in Sect. 5.2, with minor adaptations to fit the laboratory layout. The scenarios are detailed below:

5.4.1 L-shaped Hallway

The robot navigates from the bottom of a vertical corridor to the left horizontal section (Fig. 9(a-d)). As it approaches the turn, a pedestrian unexpectedly emerges from behind the corner. At the same time, another person is moving vertically from the hallway's elbow toward the bottom, walking on the left side and disregarding the keep-right convention. In this situation, the robot is expected to shift temporarily to the left side of the hallway to avoid a collision, and then take a wide curve at the corner to maintain a safe distance from the emerging pedestrian.

5.4.2 Group in Open Space

This scenario presents three people initially positioned in a circle and oriented outward (Fig. 9(e-h)). Once the robot begins navigating, the individuals turn toward each other and start interacting. The robot is expected to recognize the formation of the group and reach its goal by passing around it.

5.4.3 Non-group

This scenario is similar to the previous one, but people never turn to interact (Fig. 9(i-l)). The robot should reach its destination either by passing through the cluster or navigating around it. However, proxemics boundaries of each individual should always be respected.

5.4.4 Crowded Entrance

This scenario presents two rooms connected by two narrow entrances (Fig. 10(a-d)). As the robot approaches the closer entrance, it becomes congested with pedestrians walking through. The robot is expected to replan its path through the alternative entrance, even if this requires taking a longer route, in order to avoid cutting in front of people.

5.4.5 Queue

A queue of three people is formed with a common goal at the front of the line (Fig. 10(e-h)). The robot is expected to recognize the situation, join the queue from the end, and wait for others to advance, thus mimicking natural human behaviour and avoiding aggressive or socially inappropriate actions.

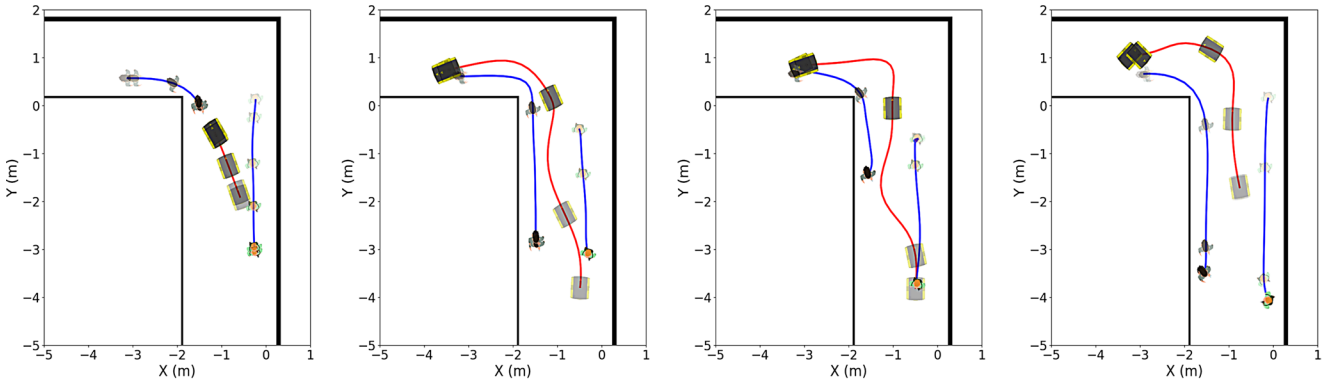
The laboratory setup, along with the Jackal robot during a queueing experiment, is shown in Fig. 11.

5.5 Real-world Results

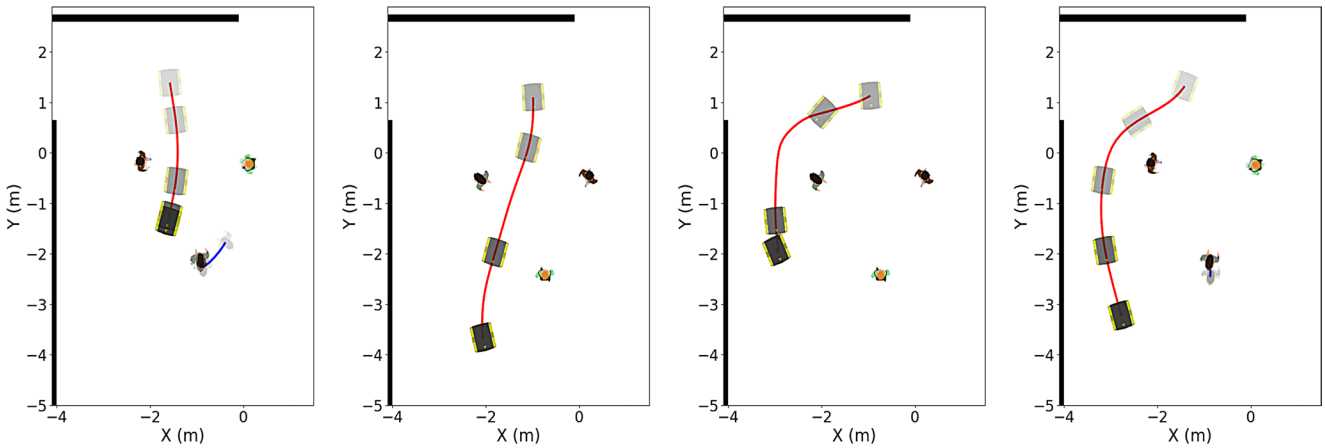
Three runs for each scenario and each algorithm have been performed. Similarly to the simulation case, the main outcomes of the experimental session are summarized in Table 3, reporting the success rate of the navigation in terms of goal-oriented task and expected social behaviour, while Fig. 12 shows the proxemics results, obtained by averaging the outcomes across all trials. In this case, the Games–Howell ANOVA analysis was not conducted on the proxemics results because the small number of samples would have resulted in limited statistical significance.

The robot's and humans' trajectories obtained from a run of each scenario are visualized in Figs. 9 and 10. Icons with varying levels of transparency indicate the positions of the robot and humans at different key frames, with sharper tracks representing more recent locations. Consequently, the most transparent traces correspond to the starting points, while the clearest traces indicate the final positions of each agent. Red lines refers to robot's trajectories, while blue lines corresponds to people's trajectories.

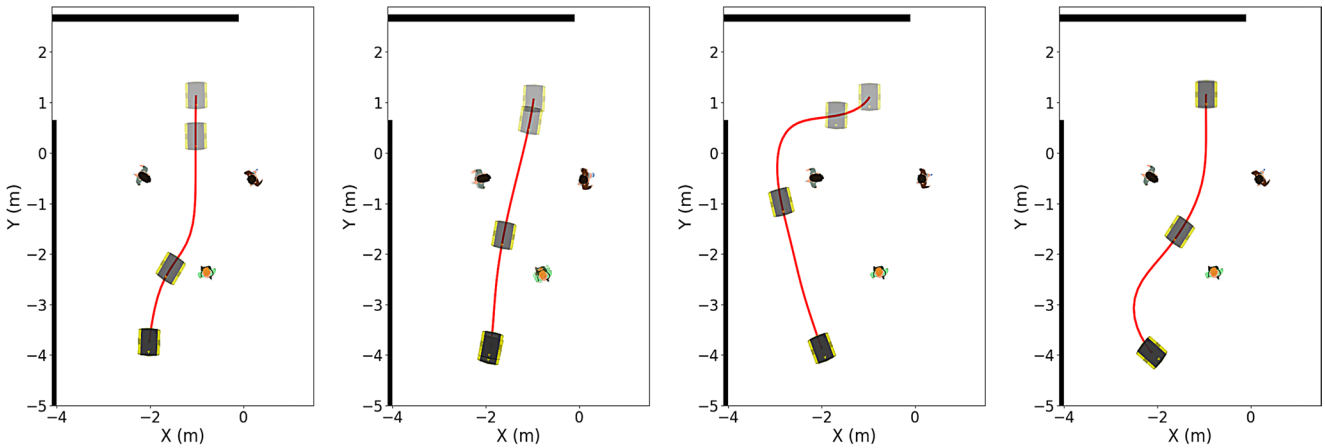
To obtain the social navigation scores reported in Fig. 13, another questionnaire was compiled right after the experiment by the volunteers involved. In this case, for each metric, participants were asked to rate the robot's behaviour on a five-point Likert scale (1–5), specifically for the scenario in which they took part. The tests involved 30 subjects, from Italy, Spain, and the United Kingdom, ranging in age from 25 to 34 years, with a mean age of 28. Approximately 27% of the participants were female and 73% male. Nearly 23% rated themselves as having low background knowledge in robotics and autonomous navigation (up to 2 out of 5), while about 70% reported high expertise (4 or 5 out of 5). The tests followed a between-subjects design: each participant took part in a subset of the scenarios, but no individual was exposed to all of them. Consequently, each participant was only presented with the survey corresponding to the specific scenario(s) in which they took part. All metrics were rated between 1 and 5 using the Likert scale, and they were finally obtained by taking the mean and the standard deviation for each metric over the total collection of answers.



(a) L Hallway - No-Layer (b) L Hallway - SNLσ_1 (c) L Hallway - SNLσ_2 (d) L Hallway - GSCL

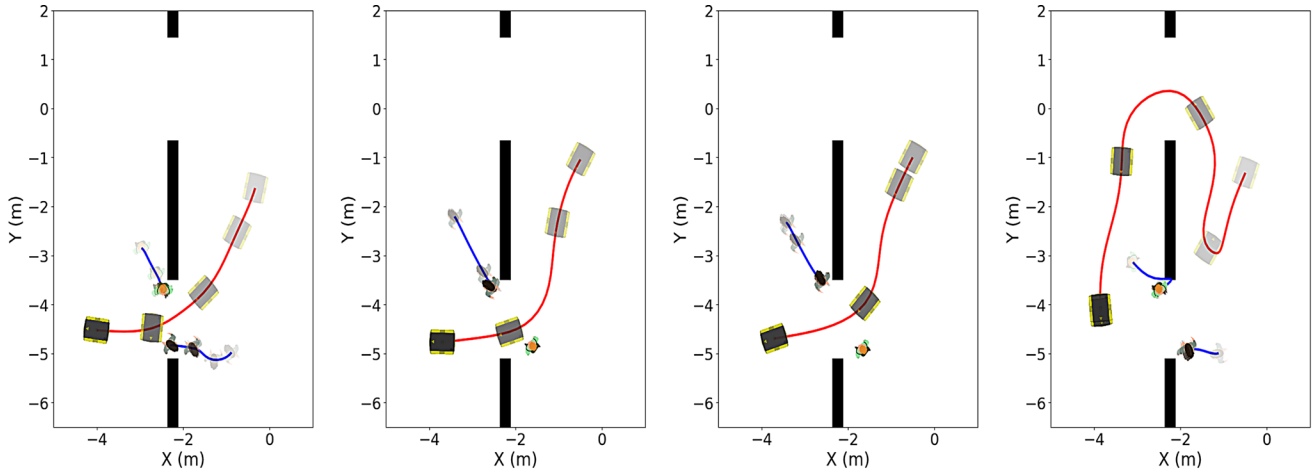


(e) Group - No-Layer (f) Group - SNLσ_1 (g) Group - SNLσ_2 (h) Group - GSCL

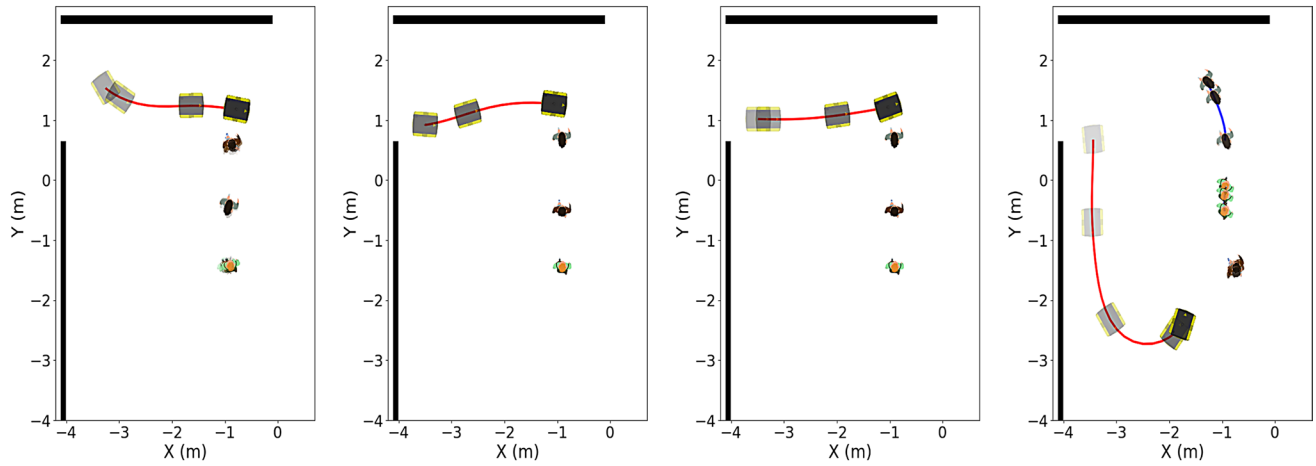


(i) Non-Group - No-Layer (j) Non-Group - SNLσ_1 (k) Non-Group - SNLσ_2 (l) Non-Group - GSCL

Fig. 9 Trajectories in the L-shaped hallway, group and non-group scenarios lab experiments comparing the proposed GSCL with the baselines. Icons with different levels of transparency are used to visualize the position of the robot and humans at different key frames. Red lines refers to robot's trajectories, while blue lines corresponds to people's trajectories



(a) Crowded entrance - No-Layer (b) Crowded entrance - SNLσ_1 (c) Crowded entrance - SNLσ_2 (d) Crowded entrance - GSCL



(e) Queue - No-Layer (f) Queue - SNLσ_1 (g) Queue - SNLσ_2 (h) Queue - GSCL

Fig. 10 Trajectories in the Crowded entrance and queue scenarios lab experiments comparing the proposed GSCL with the baselines. Icons with different levels of transparency are used to visualize the position of the robot and humans at different key frames. Red lines refers to robot's trajectories, while blue lines corresponds to people's trajectories

5.5.1 L-shaped Hallway

As expected from the simulation results, the baseline Nav-Fn planner without social layers leads the robot to collide with the person emerging from behind the corner (Fig. 9(a)). In contrast, both SNL configurations (small- and large-covariance) and the proposed GSCL enable the robot to reach the goal while avoiding collisions with people (Fig. 9(b–d)). However, as shown in Table 3, only the GSCL consistently respects the social norm under study. Specifically, with the SNL, the robot takes a very narrow turn at the corner, leaving collision avoidance with the emerging pedestrian to the controller. Conversely, the GSCL plans a wider trajectory around the corner, proactively accounting for the possible presence of a person. This advantage is also reflected in the proxemics results (Fig. 12), where the SNL and GSCL achieve similar values. However, the latter manages to spend less time in the intimate space. Moreover, in survey results (Fig. 13), the GSCL receives significantly higher scores across social compliance and safety metrics.

5.5.2 Group in Open Space

Similarly to the simulation results, the baseline Nav-Fn planner without social layers and the small-covariance SNL violate social norms by passing directly through the group (Fig. 9(e,f)). In contrast, both the large-covariance SNL and the GSCL correctly bypass the group (Fig. 9(g,h)). This outcome is confirmed by the success rates in Table 3, where all algorithms consistently reach the goal, but only the large-covariance SNL and the GSCL do so while respecting the social norm. This behaviour is reflected in the survey results



Fig. 11 Queue experiment in lab spaces

Table 3 Success rate results from the real-world experiments, reported in terms of goal-oriented task completion and adherence to expected social behaviors

Env	Method	Goal [%]	Social [%]
L Hallway	No − Layer	66.6	0
	SNLσ_1	100	0
	SNLσ_2	100	0
	GSCL	100	100
Group	No − Layer	100	0
	SNLσ_1	100	0
	SNLσ_2	100	100
	GSCL	100	100
Non Group	No − Layer	100	-
	SNLσ_1	100	-
	SNLσ_2	100	-
	GSCL	100	-
Crowded	No − Layer	100	0
Entrance	SNLσ_1	100	0
	SNLσ_2	100	0
	GSCL	100	100
Queue	No − Layer	100	0
	SNLσ_1	66.6	0
	SNLσ_2	100	0
	GSCL	100	100

The comparison covers the five considered scenarios and four navigation strategies: the baseline Nav-Fn planner, the ROS 2 SNL with low and high covariance settings, and the proposed GSCL

of Fig. 13, where the large-covariance SNL and the GSCL obtained comparable outcomes, while outperforming the other two algorithms. Similarly to the previous scenario, proxemics results for the SNL and GSCL are pretty comparable (Fig. 12).

5.5.3 Non-group

As explained in Sect. 5.3, the orientation of people does not influence the behaviour of either the Nav-Fn planner or the SNL, since the people remain stationary. Consequently, the trajectories computed by these algorithms are identical to those of the previous scenario (Fig. 9(i–k)).

On the contrary, the GSCL recognizes that the cluster of people does not constitute an interacting group and adapts its behaviour by passing through them, favouring the shortest route to the goal (Fig. 9(l)).

As in the simulation case, the quantitative results in Table 3 are of limited interest: all algorithms consistently reach the goal, and the social-based success rate is omitted since no specific social norm applies in this scenario. On the other hand, the survey results indicate that all algorithms achieved comparable scores, with the GSCL performing better than the large-covariance SNL in terms of social compliance and naturalness (Fig. 13). Regarding proxemics, the time spent in the personal space of participants is similar across all algorithms. However, the GSCL achieves the lowest time in the intimate space (Fig. 12).

5.5.4 Crowded Entrance

Consistent with the simulation results, both the SNL configurations (Fig. 10(b,c)) and the baseline Nav-Fn planner alone (Fig. 10(a)) only consider the bottom entrance in the map, since it is the shortest way. However, this choice causes the planned trajectory to cut directly in front of pedestrians using that entrance.

In contrast, the GSCL opts for a longer but more socially appropriate route by navigating through the upper, less

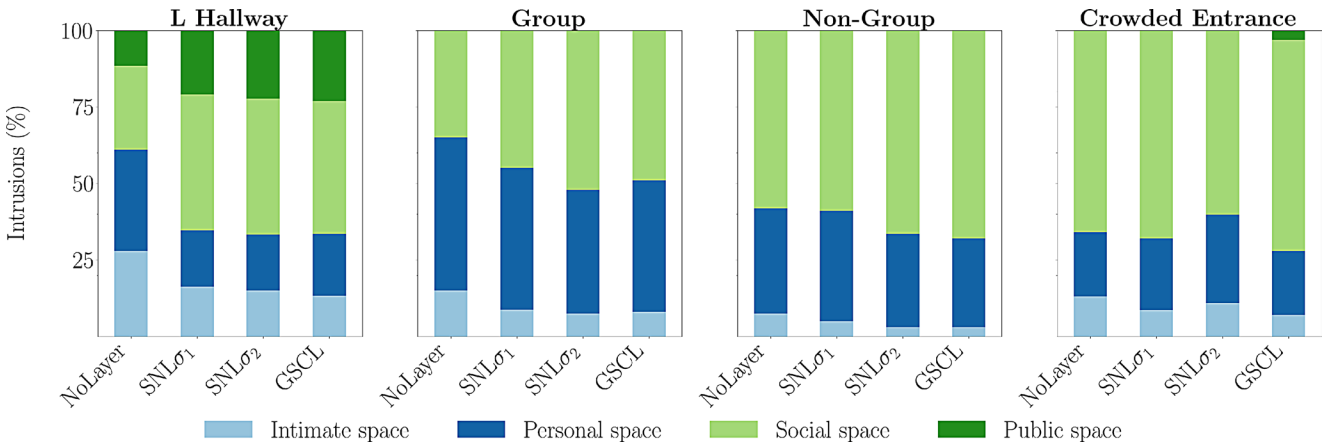


Fig. 12 Proxemics results from the real-world experiments in the four considered scenarios. The comparison includes navigation using the baselines Nav-Fn planner, the ROS 2 SNL configured with both low and high covariance values, and the proposed GSCL

crowded entrance (Fig. 10(d)). This behavior is reflected in the social-based success rate reported in Table 3, as well as in the proxemics results of Fig. 12, where the GSCL spends considerably less time in both intimate and personal spaces compared to the other algorithms.

The survey results in Fig. 13 further highlight the advantages of GSCL, showing substantially higher scores in terms of social compliance and safety. Interestingly, the GSCL's choice of a longer, socially compliant path is perceived as slightly less predictable than the more direct trajectories of the other methods, but it still achieves sightly better naturalness score.

5.5.5 Queue

Also in this case, results are comparable to those obtained in simulation. The trajectories planned by the baseline Nav-Fn planner and both SNL configurations (small- and large-covariance) lead the robot directly to the goal, cutting in front of people and violating the queueing norm (Fig. 10(e-g)). As discussed in Sect. 5.3, this outcome is unsurprising, since without a dedicated mechanism the Nav-Fn planner is unable to detect the presence of a queue or follow it appropriately.

Conversely, the GSCL correctly recognizes the queue and guides the robot at the end of the line, waiting for its turn. This result is reflected in the social-based success rate reported in Table 3, and further supported by the survey results in Fig. 13, where the GSCL achieves significantly higher scores across all four metrics compared to the other algorithms. Proxemics results are omitted for this scenario, as explained in Sect. 5.3.

6 Conclusion and Future Works

In this work, we proposed a learning-driven framework that enhances existing navigation planners by incorporating social awareness into the decision-making process. By leveraging the strengths of classical grid-based planners and integrating a flexible learning-based social cost, our approach effectively ensures practical obstacle avoidance and adherence to social norms. Unlike end-to-end learning methods, which often struggle with generalization, our framework preserves the default planner behaviour in standard navigation while introducing an adaptive and general cost layer specifically designed to handle social interactions.

Our method employs a neural network to generate a social cost map using input data that includes static obstacles, human positions, and the robot's destination. This solution allows for the integration of social requirements into classical 2D navigation, turning complex human-robot interactions into a general and well-defined planning problem.

Through extensive experimentation in both simulated and real-world settings, we demonstrated that our approach effectively generalizes across various social scenarios, such as queues, group conversations, corridor traversing, and navigating through occluded entrances. Moreover, by incorporating geometric features of the environment, we addressed additional challenges, such as traversing corridors and navigating blind corners, which further enhanced the system's adaptability.

Future research will aim to refine the learning model using richer datasets and real-time adaptation mechanisms to improve its responsiveness to unpredictable social behaviours. Another promising direction is the integration of explicit communication methods, such as gestures or verbal interactions, to enhance human-robot collaboration. Finally, evaluating our framework in long-term deployments within

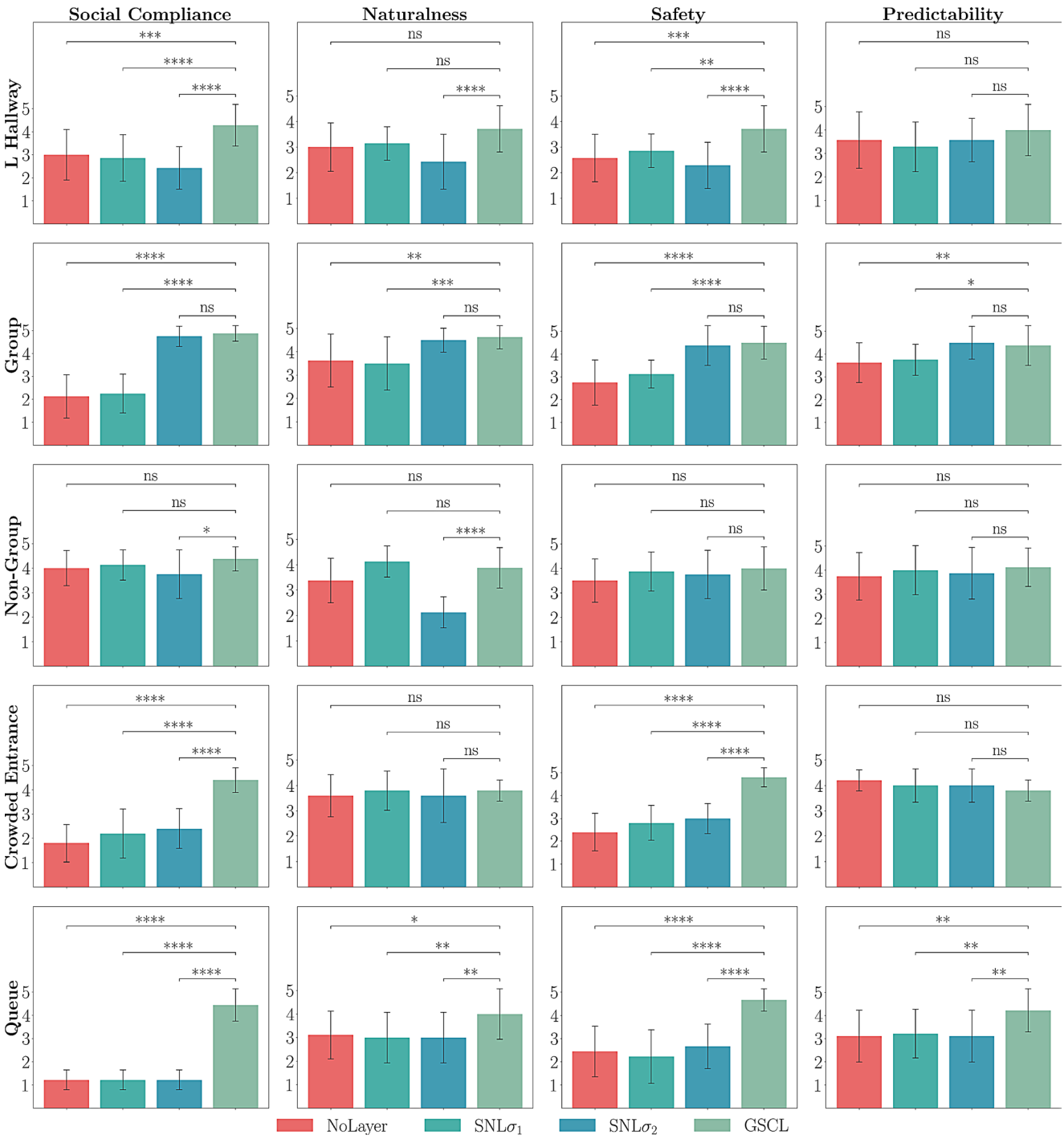


Fig. 13 Real-world experiments evaluation scores collected from the survey. Cyan columns represent the scores assigned to the navigation obtained with the social cost layer, whilst red columns represent the standard path planner, Nav-Fn. Mean values and standard deviation from the different experimental runs are reported, demonstrating the overall improvement in terms of social behaviour of the robot. The asterisks (*) indicate the p-values obtained after performing the pair-wise Tukey HSD test

real-world applications will provide deeper insights into its robustness and usability in practical settings.

Acknowledgements This work has been developed with the contribution of the Politecnico di Torino Interdepartmental Centre for Service Robotics (PIC4SeR[1]). This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 629/2024. The research has been partially funded by ISPIC (Intesa San Paolo Innovation Center) and by the SWIch action (P.R. F.E.S.R.2021/27 - D.G.R. n.19–6962) within the EMPATHY project.

[1] https://pic4ser.polito.it/.

Author Contributions Not applicable. If any of the sections are not relevant to your manuscript, please include the heading and write "Not applicable" for that section.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement. Not applicable.

Data Availability Data will be open to public upon manuscript acceptance in a dedicated repository.

Materials Availability Not applicable.

Code Availability Code will be open to public upon manuscript acceptance in a dedicated repository.

Declarations

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication Consent is provided by the author.

Competing Interests The authors declare no conflict of interest or competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Eirale A, Martini M, Chiaberge M (2025) Human following and guidance by autonomous mobile robots: a comprehensive review. IEEE Access. 13:42214–42253. https://doi.org/10.1109/ACCESS.2025.3548134
2. Eirale A, Martini M, Tagliavini L, Gandini D, Chiaberge M, Quaglia G (2022) Marvin: an innovative omni-directional robotic assistant for domestic environments. Sensors 22(14):5261
3. Triebel R, Arras K, Alami R, Beyer L, Breuers S, Chatila R, Chetouani M, Cremers D, Evers V, Fiore M et al (2016) Spencer: a socially aware service robot for passenger guidance and help in busy airports. In Field and Service Robotics: Results of the 10th International Conference, Springer, pp 607–622
4. Mavrogiannis C, Baldini F, Wang A, Zhao D, Trautman P, Steinfeld A, Oh J (2023) Core challenges of social robot navigation: a survey. ACM Trans Hum Rob Interact 12(3):1–39
5. Helbing D, Molnar P (1995) Social force model for pedestrian dynamics. Phys Rev E E 51(5):4282
6. Mavrogiannis CI, Knepper RA (2019) Multi-agent path topology in support of socially competent navigation planning. Int J Robot Res 38(2–3):338–356
7. Eirale A, Leonetti M, Chiaberge M (2024) Learning social cost functions for human-aware path planning. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 5364–5371. IEEE
8. Banisetty SB, Feil-Seifer D (2018) Towards a unified planner for socially-aware navigation. arXiv preprint arXiv:1810.00966
9. Samarakoon SBP, Muthugala MVJ, Jayasekara ABP (2022) A review on human–robot proxemics. Electronics 11(16):2490
10. Kollmitz M, Hsiao K, Gaa J, Burgard W (2015) Time dependent planning on a layered social cost map for human-aware robot navigation. In 2015 European Conference on Mobile Robots (ECMR), IEEE, pp 1–6
11. Truong X-T, Ngo T-D (2016) Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. Int J Soc Robot 8:663–684
12. Mateus A, Ribeiro D, Miraldo P, Nascimento JC (2019) Efficient and robust pedestrian detection using deep learning for human-aware navigation. Robot Auton Syst 113:23–37
13. Fang F, Shi M, Qian K, Zhou B, Gan Y (2020) A human-aware navigation method for social robot based on multi-layer cost map. Int J Intell Robot Appl 4:308–318
14. Mohamed A, Qian K, Elhoseiny M, Claudel C (2020) Social-stgcnn: a social spatio-temporal graph convolutional neural network for human trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 14424–14432
15. Zhao D, Oh J (2020) Noticing motion patterns: a temporal cnn with a novel convolution operator for human trajectory prediction. IEEE Robot Automation Lett 6(2):628–634
16. Gupta A, Johnson J, Fei-Fei L, Savarese S, Alahi A (2018) Social gan: socially acceptable trajectories with generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2255–2264
17. Vemula A, Muelling K, Oh J (2018) Social attention: modeling attention in human crowds. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp 4601–4607. IEEE
18. Mavrogiannis CI, Knepper RA (2020) Multi-agent trajectory prediction and generation with topological invariants enforced by hamiltonian dynamics. In Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13, Springer, pp 744–761
19. Yang C-T, Zhang T, Chen L-P, Fu L-C (2019) Socially-aware navigation of omnidirectional mobile robot with extended social force model in multi-human environment. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp 1963–1968. IEEE
20. Repiso E, Garrell A, Sanfeliu A (2018) Robot approaching and engaging people in a human-robot companion framework. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 8200–8205. https://doi.org/10.1109/IROS.2018.8594149
21. Repiso E, Garrell A, Sanfeliu A (2024) Adaptive social planner to accompany people in real-life dynamic environments. Int J Soc Robot 16(6):1189–1221
22. Shin H, Yoon S-E (2020) Optimization-based path planning for person following using following field. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 11352–11359. https://doi.org/10.1109/IROS45743.2020.9341637
23. Chen YF, Liu M, Everett M, How JP (2017) Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp 285–292. IEEE
24. Everett M, Chen YF, How JP (2018) Motion planning among dynamic, decision-making agents with deep reinforcement learning. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3052–3059. IEEE

25. Tai L, Zhang J, Liu M, Burgard W (2018) Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp 1111–1117. IEEE
26. Chen C, Liu Y, Kreiss S, Alahi A (2019) Crowd-robot interaction: crowd-aware robot navigation with attention-based deep reinforcement learning. In 2019 International Conference on Robotics and Automation (ICRA), IEEE, pp 6015–6022
27. Xiao X, Zhang T, Choromanski KM, Lee T-WE, Francis A, Varley J, Tu S, Singh S, Xu P, Xia F, Persson SM, Kalashnikov D, Takayama L, Frostig R, Tan J, Parada C, Sindhwani V (2023) Learning model predictive controllers with real-time attention for real-world navigation. In: Liu K, Kulic D, Ichnowski J (eds) Proceedings of The 6th Conference on Robot Learning. Proceedings of Machine Learning Research, vol 205. pp 1708–1721. PMLR, https://proceedings.mlr.press/v205/xiao23a.html
28. Choromanski K, Likhosherstov V, Dohan D, Song X, Gane A, Sarlos T, Hawkins P, Davis J, Mohiuddin A, Kaiser L, et al (2020) Rethinking attention with performers. arXiv preprint arXiv:2009.14794
29. Martini M, Pérez-Higueras N, Ostuni A, Chiaberge M, Caballero F, Merino L (2024) Adaptive social force window planner with reinforcement learning. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4816–4822. https://doi.org/10.1109/IROS58592.2024.10802383
30. Macenski S, Martin F, White R, Ginés Clavero J (2020) The marathon 2: a navigation system. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
31. Social navigation layers ROS1. https://wiki.ros.org/social_navigation_layers. Accessed 14 March 2025
32. Kendon A (1990) Conducting interaction: patterns of behavior in focused encounters, vol 7. CUP Archive
33. Marshall P, Rogers Y, Pantidi N (2011) Using f-formations to analyse spatial patterns of interaction in physical environments. In Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work, pp 445–454
34. Macenski S, Foote T, Gerkey B, Lalancette C, Woodall W (2022) Robot operating system 2: design, architecture, and uses in the wild. Sci Robot 7(66):6074. https://doi.org/10.1126/scirobotics.abm6074
35. Hall ET, Hall ET (1966) The hidden dimension 609. Anchor
36. Pérez-Higueras N, Otero R, Caballero F, Merino L (2023) Hunavsim: a ros 2 human navigation simulator for benchmarking human-aware robot navigation. IEEE Robot Automation Lett 8(11):7130–7137. https://doi.org/10.1109/LRA.2023.3316072
37. GraphPad Statistics Guide. Available at: https://www.graphpad.com/guides/prism/8/pdf/Prism-8-Statistics-Guide.pdf. Accessed 16 Dec 2025

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.