International Journal of Social Robotics
https://doi.org/10.1007/s12369-024-01190-6

SNGNN2D-v2: A GNN-Based Model for the Generation of Human-Aware Cost Maps in Dynamic Environments

Daniel Rodriguez-Criado[1] · Pilar Bachiller-Burgos[2] · Luis V. Calderita[3] · Luis J. Manso[1]

Accepted: 18 October 2024

Abstract
Navigating dynamic, human-populated environments is a critical challenge for mobile robots, as they must balance effective pathfinding with minimizing social disruption. Cost maps can combine information from different nature and are more interpretable than final control signals. This paper addresses the generation of real-time cost maps in human-aware navigation (HAN) by introducing SNGNN2D-v2, a graph neural network designed and trained to capture social interactions and respond to dynamic elements in human-populated environments. SNGNN2D-v2 is evaluated through three types of experiments. The first involves deploying a real robot in a controlled indoor environment and assessing the disturbance caused by the robot when driven by the model. The second experiment tests the proposed model under more complex and unfavorable conditions using simulated environments. Both experiments include a comparison with other proposals using social and navigation metrics. The third experiment compares SNGNN2D-v2 with an end-to-end CNN-based method to evaluate how models generalize across changes in the appearance of the environment and its elements. The results from these experiments suggest that SNGNN2D-v2 is an effective model for human-aware cost map generation for dynamic environments. Its ability to capture dynamic information, generalize across scenarios with different appearances, and represent social interactions could contribute to the development of human-friendly robots.

Keywords Human-aware navigation · Cost maps · Robotics · Graph neural networks

1 Introduction

With the increasing prevalence of smart environments, mobile robots are becoming part of our society [1–3]. Regardless of whether the task a robot performs is collaborative (e.g., a robot and a human working together), assistive (e.g., a robot accompanying a person), or independent (e.g., delivery) [4], it has to coexist in an environment with humans. To increase their acceptability, these robots must avoid disturbing humans, which requires following social conventions and behaving predictably [5]. Human-Aware Navigation (HAN) plays an important role in this goal, which requires robots to not only identify humans as dynamic entities but also consider their interactions with other individuals and objects in the surrounding environment, their intentions and their comfort [6].

The work presented in this paper is the continuation of a series of models for the estimation of the discomfort caused by a robot in the place where it operates. The first work in this series introduced *Social Navigation Graph Neural Network* (**SNGNN-v1**), presented in [7] to estimate the disturbance caused by a robot in a given scenario. This model was trained against the **SocNav1** dataset [8], which provides multiple scores for 9280 static social navigation scenarios with humans and objects. These scenarios contain human-to-human, and human-object interactions, and scores indicating

✉ Pilar Bachiller-Burgos
pilarb@unex.es

Daniel Rodriguez-Criado
danirocri@gmail.com

Luis V. Calderita
lvcalderita@unex.es

Luis J. Manso
l.manso@aston.ac.uk

[1] College of Engineering and Physical Sciences, Aston University, Birmingham, UK

[2] Tecnología de los Computadores y de las Comunicaciones, Escuela Politécnica, Universidad de Extremadura, Avenida de la Universidad, S/N, Cáceres 10003, Spain

[3] Ingeniería de Sistemas Informáticos y Telemáticos, Centro Universitario de Mérida, Universidad de Extremadura, Avenida Santa Teresa de Jornet, 38, Mérida 06800, Spain

Published online: 06 December 2024

how much the robot is disturbing. SNGNN-v1 could be used to generate disruption or cost maps for HAN by querying the network for every possible position of the robot in the room. Due to the elevated amount of queries, the time spent on generating these maps is impractical for real-time applications. Thus, SNGNN-v1 was then used in [9] to bootstrap a 2D dataset to train a new model called **SNGNN2D-v1** using a GNN-CNN combination that can generate these maps in milliseconds making it usable in real applications.

Despite the ability of SNGNN2D-v1 to generate real-time cost maps, it presents several limitations. Some of them stem from the static nature of the dataset used for training, as velocities are highly informative for social navigation and path prediction. Moreover, the interactions between entities indicate connection but do not provide any semantic information (e.g., two humans talking, humans walking together, humans shaking hands). Semantic information enriches the input and therefore the features of the GNN, potentially yielding more accurate results. Furthermore, SNGNN-v1 and SNGNN2D-v1 can only process static data and therefore their predictions only consider the information of a single instant. This fact restricts the ability of the model to make decisions based on the real-time evolution of the robot's surroundings.

The mentioned limitations motivated the development of the **SocNav2** dataset [10], comprising brief videos of a 3D environment that integrates the velocities of entities within the room. In addition to the dynamic context, SocNav2 offers several improvements over its predecessor SocNav1, such as an extended scoring system that accounts for robot movements and objectives, as well as more realistic scenarios and interactions. SNGNN-v2 was created also in [10] for generating discomfort scores from dynamic scenarios, serving as the second version of the static model SNGNN-v1.

SNGNN-v2 adopts a similar approach to its predecessor but with additional enhancements. It considers two distinct scores to assess different facets of social navigation, and the model is trained utilising dynamic scenes in which humans and the robot are in motion, addressing the primary limitation of SNGNN-v1. Again, SNGNN-v2 could be used to generate discomfort maps for HAN by querying the model for different robot positions but the generation time is too high for real-time applications. In this paper, SNGNN-v2 is employed similarly to the previous version to bootstrap a new dataset of images, used for training a new model that directly generates disruption maps, referred to as **SNGNN2D-v2**, which constitutes the second version of SNGNN2D-v1 and is the **primary contribution** of the present work.

The mentioned previous works examined two of the three most salient advantages of GNNs in the HAN domain: Firstly, GNNs permit a flexible number of input features, which proves practical in applications where there is a variable number of entities, such as humans and objects. Secondly, as GNNs can accept a graph as input, they capitalize on the relationships between entities, which are represented as edges connecting nodes in the graph. This explicit consideration of interactions results in the model attaining a more comprehensive understanding of the human-human and human-object interactions within the room. In this paper, among other contributions, we explore a third advantage, which is that the direct utilisation of structured data introduces an additional degree of abstraction. This supplementary abstraction layer allows the model to be trained on data that disregards the appearance information of the environment. Additionally, the required dataset is much smaller than what would be needed for an end-to-end solution.

To test the proposed model and show its benefits, this work presents three types of experiments. The first type of experiment (Sect. 4.2) tests the cost map generation model with a real robot in an indoor environment and compares the results with the work in [11] based on GMMs. The map is utilized by the robot's navigation planner to follow a safe path adhering to social conventions. The second type of experiment (Sect. 4.3) compares SNGNN2D-v2 against its previous version (SNGNN2D-v1), the previously mentioned GMM-based approach, and ORCA [12] in a simulated environment designed to simulate complex situations. This section also presents the results of a survey gathering users' opinions. Finally, the third type of experiment (Sect. 4.4) compares SNGNN2D-v2 with an end-to-end CNN-based method for the same task of cost map generation. The results are obtained using the same training dataset for both models.

The remaining of this work is organized as follows. Section 2 delves into the existing research on HAN in dynamic environments and highlights the gaps our proposal aims to address. SNGNN2D-v2 is described in Sect. 3, detailing every step of its development, from data acquisition to the architecture description. In Sect. 4, we present the results of the previously described experiments. Finally, section 5 summarizes the main conclusions of our work and suggests future research directions.

2 Related Work

This section expands upon the literature review in [9], focusing on approaches for addressing HAN in dynamic scenarios through the use of cost maps. Recent surveys on HAN highlight the advantages of employing maps for robotic navigation [6, 13], particularly in the context of SLAM (Simultaneous Localization and Mapping) systems, which are widely implemented in contemporary commercial robots.

The review presented in [13] categorizes cost maps used in HAN into three types: metric, semantic, and social maps. The authors acknowledge that distinctions among these categories are often ambiguous within the literature. Nevertheless, a more pronounced distinction exists between metric

maps and the other two types. Metric maps focus exclusively on the geometric aspects of the environment, whereas semantic and social maps introduce semantic information, providing an additional layer of abstraction. Besides the advantages discussed in the introduction (Sect. 1), the extra layer of abstraction offered by the GNNs also facilitates the transfer from simulation to real-world scenarios [6].

Numerous studies have explored the generation of cost maps that take into account the velocities and dynamics of environments for human-aware navigation. For instance, works such as [14, 15] extend the application of Gaussian Mixture Models (GMMs) to model areas of disruption in dynamic environments, considering the velocities of pedestrians. However, algorithms that rely on handcrafted social constraints, such as these, encounter several limitations as outlined in [9]. Specifically, they require considerable resources to develop and are difficult to debug, often leading to omitting significant variables. Additionally, these algorithms tend to oversimplify interactions and make simplistic assumptions.

The work in [16] puts forward a method for semantic robot localisation using spatio-temporal classification. The process begins with spatial classification, wherein the input is partitioned into a grid, with each grid cell assigned a label such as asphalt, cobblestones, grass, or gravel. Following this, the method's temporal aspect utilizes visual odometry to merge the derived maps. The labelled maps are subsequently projected onto the grid, and a probabilistic criterion refines the grid labels by considering neighboring cells. It is important to note that these maps treat people as mere dynamic obstacles, without incorporating relational or other semantic information. Furthermore, occupancy grids are known to have inherent limitations, including resolution constraints, memory consumption, and computational complexity.

In [17], the authors first model personal space and group interaction as social costs based on pedestrian perception, subsequently generating multi-layer dynamic cost maps. These maps incorporate social costs at various timesteps, derived from pedestrian trajectory predictions, which provide social constraints for global path planning. The global path planner then searches for the optimal state using a heuristic cost function based on the multi-layer dynamic cost maps.

In [18], the authors propose a HAN system that integrates research findings on human detection, social behavioral models, and behavior prediction. It addresses social distance considerations, consolidating information into a dedicated layer for human behavior intention cognition. The trajectory is then optimized using a dynamic triangular window method that incorporates human behavioral intention cognition, ultimately determining a suitable robot trajectory. The main limitation of these approaches is disruption areas are modelled with analytic functions, which are more susceptible to errors and inconsistencies than a DL-based approach. Moreover, some social aspects are difficult to express analytically. For instance, the density of people in the environment may play a crucial role in the interpretation of disruption. In crowded spaces, the discomfort area of humans tends to narrow compared to scenarios with less dense spaces [10].

Alternative approaches for generating cost maps that consider environmental dynamics utilize graph structures. [19] propose a semantic framework that models the environment based on natural language descriptions and scene classifications. The topology of the resulting graph contains nodes representing the robot's trajectory, while the edges indicate connectivity between the nodes. The temporal component is involved in updating the graph topology by taking into account previous metric exteroceptive sensor data, scene appearance observations, and natural language descriptions. Similarly, [20] capitalizes on the temporal component, introducing a time-evolving navigation graph that delivers a semantic topology of the explored area and the connectivity among detected places in terms of inter-place transition probability.

Both aforementioned works demonstrate the advantages of employing graphs for incorporating semantic information into the model and explicitly accounting for element interactions. However, it is important to note that these studies do not specifically address the HAN problem, as they do not consider humans as entities in their navigation models.

Finally, it is worth mentioning that cost maps can also be created from visual features using end-to-end deep learning models for image generation [21, 22]. The main limitations of these models are their adaptability constraints to new scenarios and poor performance in modelling interactions between entities in the environment. Owing to Generative Adversarial Networks' (GAN) ability to generate high-resolution images relatively quickly, a state-of-the-art GAN model is selected for comparison with SNGNN2D-v2 in Sect. 4.4. Specifically, the model presented in [23], also known as *Pix2pix*, is chosen for this purpose.

3 Method

The previous version of our proposal, SNGNN2D-v1, leverages GNNs to provide human-aware 2D cost maps for robot navigation. While the model showed good performance for certain scenarios, its effectiveness in dynamic environments was constrained by the following limitations:

- It only considered static features of the entities in a scenario (e.g., the position and orientation of humans).
- The interactions between entities were represented without incorporating semantic information (there is no distinction between two standing people talking and two people walking together).

- The model input represented only a single moment in time, lacking temporal context.

SNGNN2D-v2 emerges as the solution to overcome these limitations, offering a practical approach to human-aware navigation in dynamic and complex scenarios.

While the nature of both models differs, their creation strategy follows a common set of steps:

1. Create a single-output model that estimates the discomfort of humans in the robot's presence;
2. Generate a discomfort map dataset by performing multiple queries to the aforementioned model;
3. Use the generated dataset to train a model that generates maps instead of scalars.

The subsequent sections describe these steps in detail.

3.1 Single-Output Model for Discomfort Estimation in Dynamic Scenarios

Building sufficiently large and diverse datasets has become one of the main challenges in deep learning. Data should ideally cover various scenarios and edge cases. Noisy or biased data can mislead models, so ensuring data quality is crucial. For supervised learning, data needs to be labelled or annotated, which can be a time-consuming and expensive process. Annotation errors and other inconsistencies can adversely affect model training. In the context of human-aware navigation, the complexity of the process of annotating disruption or cost maps from navigation scenarios makes these challenges especially difficult to solve. Our initial proposal for this problem was to create a dataset that associates each scenario with a single discomfort score. In this paper, such a dataset (**SocNav2**) is extended to create a 2-dimensional dataset, as will be explained in section 3.2.

The scenarios compiled in SocNav2 were generated using *SONATA* [24], a tool designed to simulate dynamic human-populated navigation scenarios. While SONATA exclusively provides simulated scenarios, the use of synthetic data is crucial in the context of human-aware navigation. This is primarily because generating a comparable number of scenarios using only real-world data would be infeasible. Furthermore, situations that jeopardize human safety, such as human-robot collisions, cannot be ethically performed in real-world settings.

Each sample of SocNav2 consists of 35 "snapshots" of a scene of a room with a moving robot, a goal position for the robot, objects, and potentially moving humans, taken during a time interval of a few seconds (see Fig. 1). Humans may interact with other humans or objects in the room. The annotation of the data corresponds to the scores for two social navigation-related statements: "*the robot does not cause any disturbance to the humans in the room*" (***Q1***) and "*the robot is moving towards the goal efficiently, not causing any disturbance to the humans in the room*" (***Q2***). The scores range from 0 to 100 to represent situations that go from *unacceptable* to *perfect*.

Six subjects participated in scoring the dataset, yielding a total of 13, 406 scored samples. This initial set was extended using a process of data augmentation that resulted in a final dataset comprising 53, 600 samples. More details about the dataset and its generation can be found in [10].

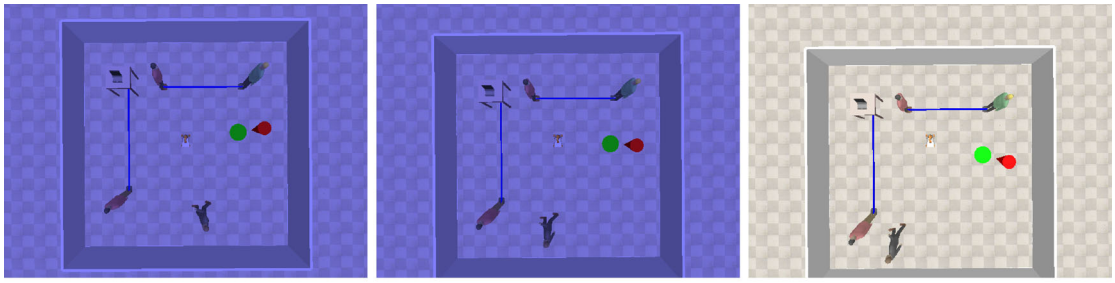
Using SocNav2 and GNNs, a model to predict discomfort scores in dynamic scenarios was developed [10]. This model, named SNGNN-v2, receives a graph representing the scenario through time as input and produces two values corresponding to the scores for ***Q1*** and ***Q2***. The input graphs are composed of a sequence of three sub-graphs corresponding to three snapshots of the videos shown to the subjects. Each sub-graph (referred to as a 'frame graph') is separated by a one-second interval. The graph creation process entails two steps. First, each snapshot is transformed into a separate frame graph. Entities in the scene are represented as separate nodes, considering 5 types of nodes:

- **room (r):** There is one room node per frame graph. It acts as a global node and is also used to include the information of the robot.
- **wall (w):** A node for each of the segments defining the room limits.
- **goal (g):** Used to represent the position that the robot must reach.
- **object (o):** A node for each object in the scenario.
- **human (h):** A node for each human.

The room node is connected in both directions to any other node of the graph for that frame. Using a global node favors communication across the graph and reduces the number of layers required [25]. In addition, for every human involved in interactions, two new edges are added between the human and the entity (human or object) they interact with, one in each direction. The graphs also include self-edges for all nodes.

Once the three frame graphs in the sequence have been generated, they are merged into a single graph representing the sequence (see Fig. 2). This temporal connection is established with an edge linking the node in each frame graph to the corresponding node in the subsequent frame graph.

The feature vectors of the nodes are constructed by concatenating several sections. The first two sections consist of one-hot encodings that specify the node types and the frame to which they belong. The remaining sections are type-specific and contain data only if the node matches that type; otherwise, they are filled with zeros. For human, wall, and object nodes, the features in these sections include position,



(a) First second. (b) Second second. (c) Third second

Fig. 1 Three snapshots of a *SocNav2* dataset sample

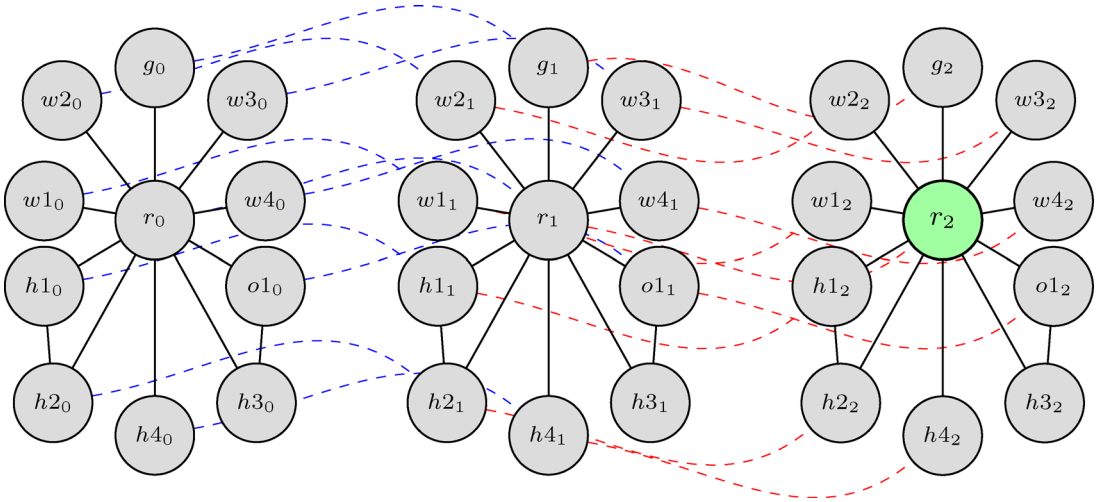


Fig. 2 Example of how the scenario-to-graph transformation works, based on the scenario depicted in Fig. 1. The sub-indexes of the node names refer to the time frame of each node

distance to the robot, speed, and orientation, all referenced to the robot's frame. For normalization purposes, position and distance are represented in decametres. Similarly, orientation is represented as sine and cosine components, rather than the angle itself. This orientation representation enforces periodic rotational equivariance with a period 2π radians, avoiding problems related to the use of different values for representing the same or close angles (for instance, 2π and 0). In the case of wall segments, the position denotes the segment's centre, and orientation represents the tangent. Object sections additionally include width and height features that define the object's bounding box. For nodes corresponding to room symbols, the section includes the normalized number of humans in the room and the robot's velocity command. An illustrative layout of this structure can be found in Table 1.

Edge features are dependent on the type of GNN block used to create the network architecture. Thus, edges do not include any information in graph convolution networks (GCN) or graph attention networks (GAT). However, relational graph convolutional networks (R-GCN) blocks support edge labels. For this kind of block, a different label is used for each possible type of relation (e.g., human-human, human-room, wall-room). Finally, for message passing neural networks (MPNN), edges may include a vector feature with additional information. Specifically, we have used one-hot encodings to represent the different types of relation between two nodes and one additional feature to include the distance between 2 entities.

Table 1 Structure of the feature vectors of nodes

n. one-hot	5 elements (one per node type)				
f. one-hot	3 elements (one per frame graph)				
room	number of humans		adv. speed	rot. speed	
human	position	speed	orientation	distance	
object	position	speed	orientation	distance	shape
wall	position		orientation	distance	
goal	position			distance	

To create our discomfort prediction model (SNGNN-v2), we followed a process of hyperparameter tuning, training different architectures with different GNN blocks on the SocNav2 dataset. According to the connectivity of the graph, all nodes are directly or indirectly connected to the room node

of the last frame (green node of Fig. 2). Using this fact, the GNNs were trained to perform backpropagation based on the feature vector of that node in the last layer. After more than 300 training sessions, the best-performing model was composed of a sequence of **6 MPNN blocks** with 40, 30, 21, 12, and 3 hidden units and 2 output units, corresponding to the scores for **Q1** and **Q2**.

3.2 Generation of a Discomfort Map Dataset

Training a discomfort map estimation model in a supervised fashion requires a dataset associating scenarios to such kinds of maps. SocNav2 is not suitable for that purpose as it only provides annotations for discomfort values considering that the robot is positioned at a specific location in the environment. Nevertheless, SNGNN-v2, trained using SocNav2, can produce a spectrum of discomfort values for a specific scenario through successive model queries, each time varying the robot's position in the input of the model. Figure 3 illustrates this idea.

According to this strategy, the generation of a cost map involves an iterative process, where each iteration produces a discomfort value for a specific position on the map. More precisely, given a dynamic scenario and a particular map position, we calculate the discomfort value using SNGNN-v2 to estimate the *Q1* score for that robot position in the scenario. This involves creating the input graph for the model, with the robot's position varying for each iteration, as illustrated in Fig. 3. Note that only the *Q1* score has been used since it does not account for the goal position of the robot. Besides, since the generated maps are primarily intended for planning purposes, specific robot movements are not considered. Consequently, to generate each map, the robot remains stationary while creating the graphs used as input for the SNGNN-v2 model.

Using this approach, we generated a dataset associating dynamic scenarios with discomfort maps for human-aware navigation. Each map has a resolution of 150×150. This resolution was selected to balance the quality of the resultant image against the generation time. The dataset comprises 17, 044 scenarios in total, split into 13, 600 training samples, 1, 717 for evaluation, and 1, 727 for testing.

3.3 Estimation of Discomfort Maps for Dynamics Scenarios

The process described for generating discomfort maps is impractical for real-time applications. Even on a high-performance computer, this method could take several minutes, particularly for complex scenarios. As introduced previously, to address this issue, we propose SNGNN2D-v2, a neural network capable of learning the relationship between scenarios and maps using the generated dataset.

As SNGNN2D-v2 is designed to generate maps from graph representations of scenarios, it has to integrate distinct modules associated with various neural network architectures. The initial module comprises a GNN that processes a graph representing the scenario and produces an intermediate map representation capturing essential information. In turn, the second module takes this intermediate representation and generates an image corresponding to the intended discomfort map. This task is accomplished through the utilization of a CNN.

Regarding the GNN, since the raw data considered in both models, SNGNN-v2 and SNGNN2D-v2, is the same, the scenario information in the 2D version is encoded using the same type of graph as in the scalar version. However, modifications have been made to these new graphs to optimize their generation and ensure compatibility with CNN.

Firstly, the entity graph, which encodes room information as depicted in Fig. 2 across three distinct time-frame graphs, is merged into a singular graph, omitting the goal node. Temporal information is now encoded within the nodes' feature vectors, eliminating the need for temporal connections. This restructuring leads to a simplified graph, enhancing processing speed and efficiency.

Secondly, the graph incorporates a lattice of nodes that delineate a square area surrounding the center of the frame of reference. The primary objective of this grid is to establish a direct connection between the GNN and the CNN through an intermediary representation that both modules can efficiently process. The number of nodes and the area they cover are tunable hyperparameters, balancing performance, computational time, and area coverage.

Finally, the graph resulting from the merged temporal graphs is integrated with the grid by linking each entity node to the nearest grid node, spatially. Each entity node can be connected to multiple grid nodes within a specific radius, leading to the final unified graph depicted in Fig. 4.

With the union of the three graphs into one, the features of each node representing an entity are likewise merged. The metrics section for each entity node type, excluding the features of the goal node, which is not included in the graph, is replicated three times, each corresponding to a different frame. To indicate the number of available frames, a one-hot encoding is used, ranging from 1 to a maximum of 3. In cases where a frame is unavailable, the fields corresponding to that frame are populated with zeros.

Furthermore, to accommodate the grid node type, an additional element is added to the node type's one-hot encoding, along with a concatenation of grid features. These grid features encompass the 2D position of the grid node and its distance from the center. Table 2 shows a comprehensive overview of these node features.

Regarding edge features, new labels are introduced for the grid connections. Specifically, a label is assigned for each

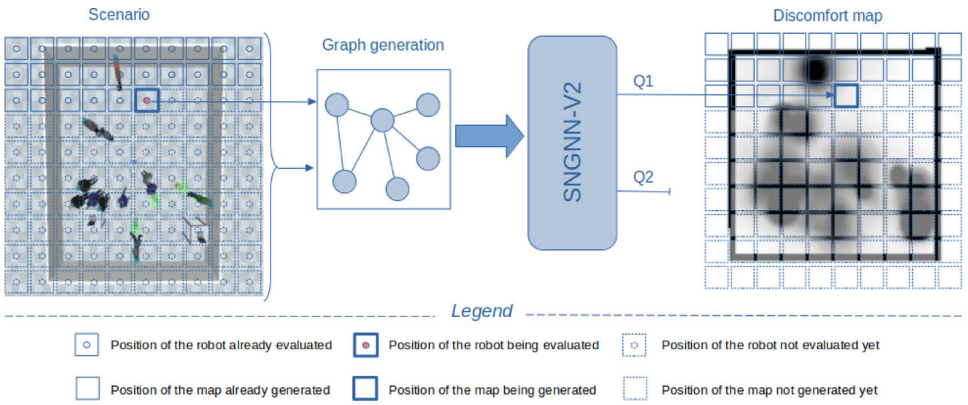


Fig. 3 Process of generating discomfort maps using SNGNN-v2

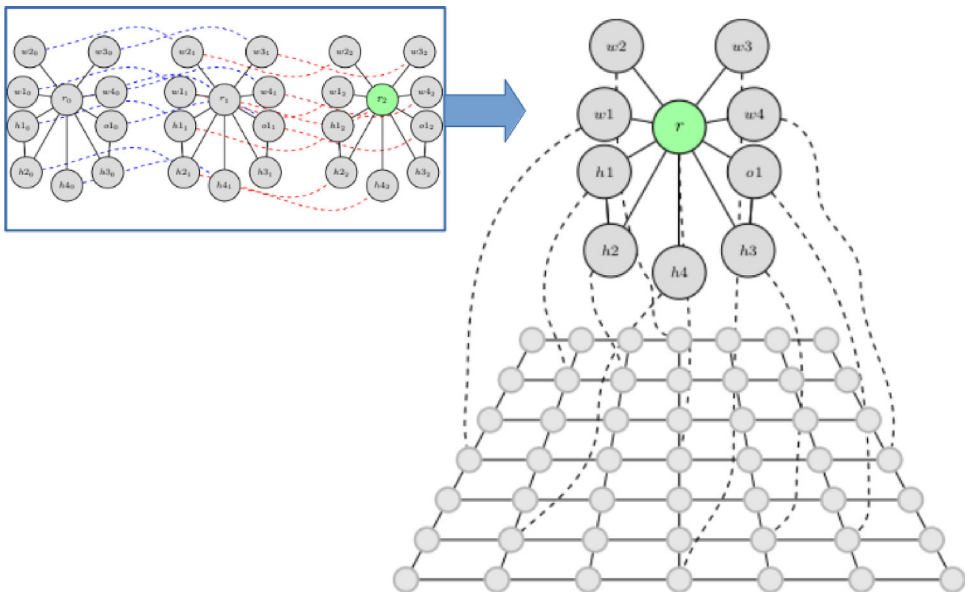


Fig. 4 Adaptation of the graph in Fig. 2 including the lattice of nodes and merging the 3 temporal graphs into a unique graph

Table 2 Features of nodes for SNGNN2D-v2

n. one-hot		5 elements (one per node type)				
f. one-hot		3 elements (max. number of frames)				
grid		position			distance	
x3	room	number of humans		adv. speed	rot. speed	
	human	position	speed	orientation	distance	
	object	position	speed	orientation	distance	shape
	wall	position		orientation	distance	

room entity connecting to the grid (e.g., wall-grid, room-grid, human-grid). Additionally, the labels of edges within the grid are differentiated based on the direction of the connection, ensuring an accurate representation of their relative positions (i.e., up, down, left, right).

The proposed architecture is depicted in Fig. 5. The input graph to the GNN is composed of the entity graph representing the room and the grid of nodes that facilitate the connection between the GNN and the CNN. Since the GNN generates an output graph with the same structure as the input graph, it includes additional nodes incompatible with the CNN. To ensure compatibility, a filtering module is responsible for retaining only the grid nodes. Each grid node can be linked to a pixel, with several channels corresponding to the node's features. Following this node-pixel analogy, an image is formed, serving as the initial representation of a disruption map for the given scenario. However, it possesses a limited resolution due to a constraint on the number of grid nodes to maintain GNN processing efficiency. To achieve a final map with the desired resolution, the CNN processes an upsampled version of the grid-to-image conversion, ultimately producing an enhanced disruption map. We specifically employ a ResNet-based architecture for the CNN, which consists of 6 ResNet blocks positioned between downsampling and upsampling layers. This architectural choice aligns with one of the potential network architectures used in *Pix2pix* [23]. Our selection is motivated by two key factors. Firstly, it has demonstrated strong performance in image transformation tasks. Secondly, it serves to illustrate the advantages of our approach compared to an end-to-end solution.

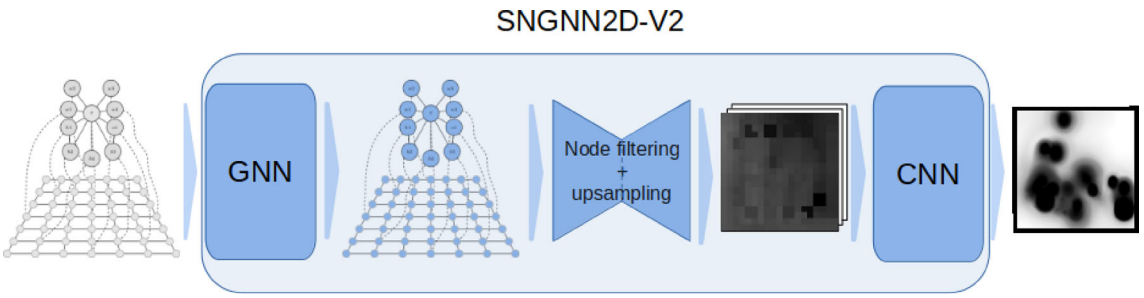


Fig. 5 SNGNN2D-v2 architecture

Table 3 Hyperparameters used to train the best model for SNGNN2D-v2

Hyperparameter	Value
Batch size	40
CNN input channels	35
Learning rate	5e−5
Activation GAT layer	elu
Final activation GAT	relu
GAT hidden units	[95, 71, 62, 57, 45, 35]
GAT heads	[34, 28, 22, 15, 13, 10]
Alpha	0.2088642

4 Experimental Results

To substantiate the proposed model, this section encompasses a comprehensive experimental procedure, addressing: **a)** the precision of the model and its time efficiency (Sect. 4.1); **b)** an assessment of its integration within the ROS navigation stack in a real robot (Sect. 4.2); **c)** A comparison of the model's previous version, a GMM based model [11], and the Optimal Reciprocal Collision Avoidance (ORCA) approach, using social metrics in simulated scenarios (Sect. 4.3); and **d)** a comparative analysis with a CNN-based approach (Sect. 4.4).

4.1 Training Results: Time and Accuracy

To find the best architecture for the proposed model, we followed a process of hyperparameters' random search using several GNN's types, numbers of layers and neurons, and activation functions. After this process, the model showing the best performance achieved an MSE of **0.001872**, **0.004833** and **0.004750** for the training, development, and test datasets, respectively.

The architecture obtaining the best results is a Graph Attention Network with 7 layers, followed by a ResNet-based CNN module as explained in Sect. 3.3.

Table 3 shows the hyperparameters used to train the best model.

Figure 6 shows the output of SNGNN2D-v2 in comparison with the ground truth from the bootstrapped dataset. As can be observed, the visual results are very similar. In addition, the average time for a query to the network, measured in an NVIDIA Jetson AGX Orin,[1] is **11** milliseconds, which makes the model suitable for real-time use.

A video showing the real-time generation of maps from scenarios created with SocNavGym [26] can be found at https://github.com/gnns4hri/SNGNN2D-v2.

4.2 Real Environment Evaluation

To evaluate the maps generated by SNGNN2D-v2 in a real scenario, we use the differential RB-1 base from Robotnik.[2] The robot employs the ROS navigation stack with a Timed Elastic Band (TEB) planner ([27]). The generated cost maps are published as a ROS topic, to which the robot's navigation system can subscribe and use as a local map for TEB. For comparative purposes, the maps generated using GMMs in the model developed by [11] were also tested using the same planner.

Although different controllers might yield varying outcomes, in our experiments, the same controller is consistently used across all methods to ensure a fair comparison. The primary focus is to evaluate the cost map generation rather than the controller itself. By keeping it constant, we isolate the impact of the cost map generation on the overall performance.

Human positions and velocities within the room are detected using the 3D pose estimator presented in [28]. A ROS plugin overlays the generated map onto the map created by the robot's laser, enabling objects to appear in the final map.

Each map was tested five times across seven different scenarios, illustrated in Fig. 7. The robot's starting and goal positions remained constant across all experiments. The description of each scenario is as follows:

[1] NVIDIA Jetson Orin specification: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/.

[2] Robotnik RB1-base specifications: https://robotnik.eu/products/mobile-robots/rb-1-base-en/.

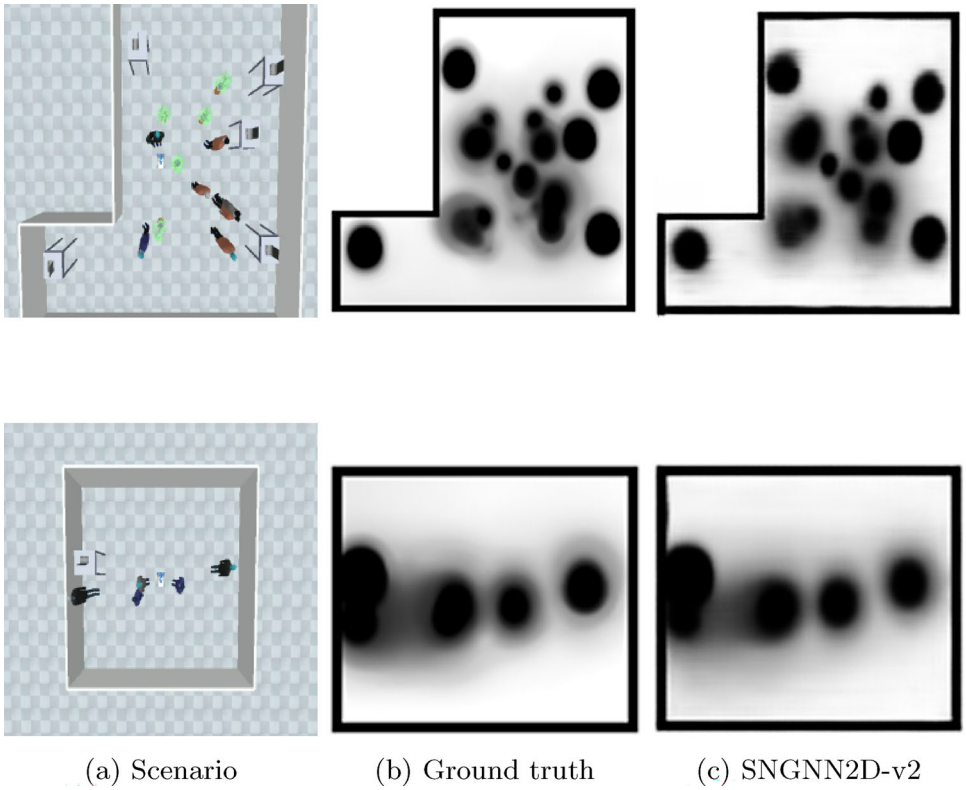


Fig. 6 Results generated by SNGNN2D-v2 (right column) compared with the ground truth (middle column) for two *SONATA* scenarios (left column). The first row shows the results for a L-shaped room and the second row for a square-shaped room

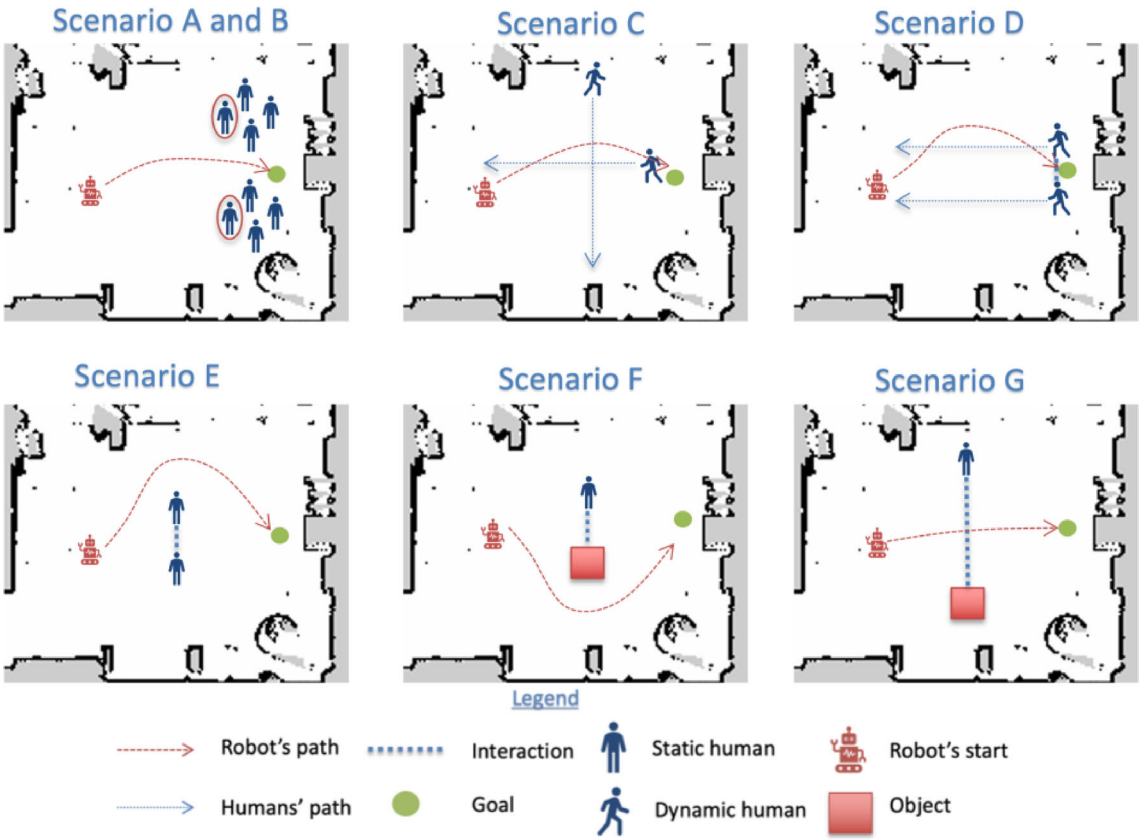


Fig. 7 Schemes of the different scenarios used to test the maps

- **Scenario A** (S_A)**:** This scenario features two groups of three stationary humans. The centres of the two groups are separated by 2.5 meters. Each member of a group is located 0.8 meters from the group centre. The robot's goal is midway between both groups, requiring the robot to traverse between them.
- **Scenario B** (S_B)**:** Identical to S_A, but with an additional human added to each group. The newly added human is highlighted in red in the top-left image of Fig. 7.
- **Scenario C** (S_C)**:** This scenario involves two moving humans: one moving opposite to the robot and another perpendicularly. Both humans start moving at the same time as the robot.
- **Scenario D** (S_D)**:** Two humans move towards the robot's initial position, side by side, with an interaction between them. The robot's goal is at the midpoint between the initial positions of the humans.
- **Scenario E** (S_E)**:** This scenario comprises two stationary humans interacting with each other. They stand 1 meter apart, facing each other as if in conversation. The robot should circumnavigate the interaction area to reach the goal on the other side.
- **Scenario F** (S_F)**:** Similar to the previous scenario, but substitutes one person for an object measuring 0.8×0.8 meters. The human faces the object at a distance of 1 meter, interacting with it. Once again, the robot should avoid the interaction area.
- **Scenario G** (S_G)**:** Identical to the previous scenario but with the human and object separated by 3 meters. Given this setup, the robot has insufficient space to bypass the person or object, so it must traverse the interaction area while minimising disruption.

It is important to mention that only scenarios C, D, and E were recorded with real people present. In the other experiments, where the people were stationary, their positions were hard-coded to generate the cost maps, thereby avoiding the small error introduced by the pose estimator in determining their positions. Additionally, it is worth noting that the resolution of the GMM maps had to be reduced by half to ensure real-time responses. The default resolution proved to be too slow for its use with the robot planner.

Figure 8 exemplifies scenarios C and E, which were recorded with real people. The right-hand images showcase the robot's path, along with the positions of the individuals tracked by the 3D pose estimator presented in [28].

The completion of the experiments was followed by the evaluation of results using the same metrics that are outlined in the experiments in [9], with an additional success rate metric:

- ***sr***: percentage of experiments in which the robot reaches the goal.

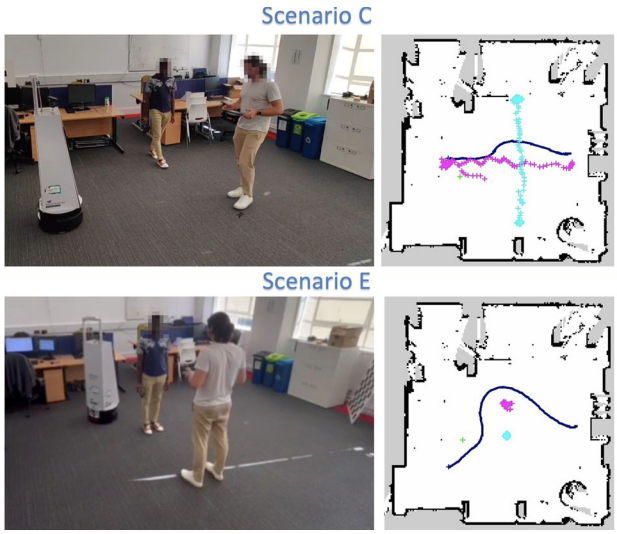


Fig. 8 Images of the experiments for scenarios C and E. The left-hand images depict footage captured during the experiments, while the right-hand images display the recorded positions of the robot and humans. The robot's path is illustrated by a dark blue line, while magenta and cyan colors represent the positions of the two individuals involved in the experiments

- $\boldsymbol{si}_i$: average percentage of intrusions into the intimate space of humans (closer than 0.45m).
- $\boldsymbol{si}_p$: average percentage of intrusions into the personal space of humans (closer than 1.2m).
- $\boldsymbol{si}_r$: average percentage of intrusions into an interaction (closer than 0.5m).
- $\bar{\boldsymbol{t}}$: average normalized time to reach the goal.
- $\overline{\boldsymbol{pl}}$: average normalized path length.
- ***CHC***: average cumulative heading changes.
- $\boldsymbol{dh}_{min}$: average minimum distance to a human.

Table 4 presents the outcomes for all the metrics across all experiments conducted using GMM and SNGNN2D-v2 maps. The best metrics values are highlighted in bold.

Both methods show comparable results across most metrics. In addition, the limited number of experiments per scenario does not allow for determining significant statistical differences for certain results. Nevertheless, one of the main differences between the two methods becomes apparent when the scenario involves dynamic elements. In such cases, the map generation time required by GMM is too slow to react effectively to moving individuals, leading to the robot failing to reach its goal. This is especially notable in Scenario D, as indicated by the success rate of both methods.

In scenarios with static groups (A and B), both methods perform similarly. However, the robot using SNGNN2D-v2 maps tends to get closer to humans as the number of people increases. In densely populated scenarios, this closer proximity may be considered acceptable due to the limited available

Table 4 Results of the metrics for each of the scenarios comparing SNGNN2D-v2 with the GMM model in [11]

		S_A	S_B	S_C	S_D	S_E	S_F	S_G
$sr(\%)$	GMM	100	100	80	40	60	100	100
	SNGNN2D-v2	100	100	80	**100**	**100**	100	100
$si_i(\%)$	GMM	0	0	0	2.963 ± 2.963	0	0	0
	SNGNN2D-v2	0	0	0	**0**	0	0	0
$si_p(\%)$	GMM	$\mathbf{59.52 \pm 3.31}$	$\mathbf{64.414 \pm 3.702}$	18.855 ± 1.741	27.164 ± 1.725	$\mathbf{10.155 \pm 3.451}$	15.365 ± 30.729	0
	SNGNN2D-v2	78.266 ± 4.722	75.216 ± 1.41	$\mathbf{13.799 \pm 5.371}$	$\mathbf{23.342 \pm 1.884}$	37.873 ± 3.345	**0**	0
$si_r(\%)$	GMM	0	0	0	0	0	0	100
	SNGNN2D-v2	0	0	0	0	0	0	100
$\bar{t}$	GMM	5.211 ± 0.486	4.437 ± 0.244	$\mathbf{4.433 \pm 0.121}$	$\mathbf{4.852 \pm 0.005}$	5.042 ± 0.192	$\mathbf{4.134 \pm 1.029}$	$\mathbf{3.383 \pm 0.231}$
	SNGNN2D-v2	$\mathbf{5.115 \pm 0.946}$	$\mathbf{4.323 \pm 0.189}$	4.718 ± 0.192	5.298 ± 0.479	$\mathbf{4.837 \pm 0.293}$	4.788 ± 0.381	3.868 ± 0.182
$\overline{pl}$	GMM	1.134 ± 0.625	1.077 ± 0.026	0.992 ± 0.013	$\mathbf{1.009 \pm 0.008}$	1.051 ± 0.057	$\mathbf{0.967 \pm 0.034}$	1.019 ± 0.044
	SNGNN2D-v2	$\mathbf{0.967 \pm 0.025}$	$\mathbf{1.01 \pm 0.016}$	$\mathbf{0.991 \pm 0.002}$	1.02 ± 0.02	$\mathbf{0.969 \pm 0.014}$	0.996 ± 0.025	$\mathbf{0.998 \pm 0.027}$
$CHC(rads)$	GMM	$\mathbf{3.574 \pm 0.627}$	$\mathbf{2.096 \pm 0.353}$	3.609 ± 0.259	$\mathbf{1.801 \pm 0.807}$	4.705 ± 0.448	$\mathbf{3.416 \pm 1.348}$	$\mathbf{1.011 \pm 0.209}$
	SNGNN2D-v2	4.57 ± 1.696	2.431 ± 0.0435	$\mathbf{3.041 \pm 0.754}$	2.623 ± 0.506	$\mathbf{4.139 \pm 0.776}$	3.773 ± 1.112	1.21 ± 0.349
$dh_{min}(m)$	GMM	0.801 ± 0.065	$\mathbf{0.816 \pm 0.019}$	0.585 ± 0.034	0.471 ± 0.083	$\mathbf{1.038 \pm 0.044}$	1.604 ± 0.443	1.688 ± 0.033
	SNGNN2D-v2	$\mathbf{0.81 \pm 0.03}$	0.731 ± 0.022	$\mathbf{0.771 \pm 0.218}$	$\mathbf{0.607 \pm 0.041}$	0.861 ± 0.048	$\mathbf{2.006 \pm 0.106}$	$\mathbf{1.878 \pm 0.045}$

space [10]. This social aspect is challenging to fully represent in a model-based approach.

Finally, in scenarios involving human-human and human-object interactions (E, F, G), the robot's performance using SNGNN2D-v2 maps is similar to that of GMM maps in terms of speed and heading changes. However, when the robot needs to navigate through these interactions (Scenario G), it tends to maintain a greater distance from humans when using SNGNN2D-v2.

While further experiments in real crowded scenarios are necessary to confirm the successful deployment of SNGNN2D-v2 on real robots, these initial results provide a promising foundation for future work.

4.3 Evaluation in Simulated Scenarios

To extend the number and variety of situations for evaluating our proposal, SNGNN2D-v2 has also been tested in simulated scenarios using SocNavGym [26]. Specifically, we present a comparison between SNGNN2D-v2, SNGNN2D-v1, GMM [11], and ORCA [12] for two different simulated scenario configurations using a set of evaluation metrics based on the ones suggested by [29]:

- ***sr***: success rate.
- ***cr***: collision rate (considering humans, objects, and walls).
- ***wcr***: wall collision rate.
- ***ocr***: object collision rate.
- ***hcr***: human collision rate.
- ***rcp***: percentage of collisions against humans caused by the robot. A collision is considered to be caused by the robot if its linear velocity is not zero at the moment of the collision.
- ***to***: percentage of failures caused by timeout before reaching the goal.
- ***fp***: average percentage of steps in the episode in which the robot does not decrease the distance to the goal.
- ***st***: average time in the episode in which the linear velocity of the robot is zero.
- ***t***: average time to reach the goal.
- ***pl***: average path length.
- ***spl***: success weighted by path length.
- ***psc***: average percentage of steps in the episode in which the robot complies with personal space.
- $\boldsymbol{dh}_{min}$: average minimum distance to human.
- $\boldsymbol{cd}_{min}, \boldsymbol{cd}_{avg}$: minimum and average distance to obstacles (on average considering all episodes).
- $\boldsymbol{v}_{min}, \boldsymbol{v}_{avg}, \boldsymbol{v}_{max}$: minimum, average and maximum linear velocity of the robot (on average considering all episodes).

The two scenario configurations used in this evaluation consist of a $10m \times 10m$ room with objects and humans in it. Each scenario includes a robot and a goal position. All entities, including the goal, are randomly located. The number of objects varies from 2 to 5. Humans can be static or dynamic and, additionally, can be interacting with another human or an object. The number of static humans, interacting or not, ranges from 0 to 2. The difference between the two configurations is in the number of dynamic humans and dynamic interactions (two humans walking together). In the first configuration, the number of dynamic humans varies between 0 and 2 and the same goes for the number of dynamic interactions. On the contrary, in the second configuration, the number of both, dynamic humans and interactions, has been set to 2. Dynamic humans do not always consider the robot in their policy. Particularly, the probability that a moving human explicitly avoids the robot has been set to 0.5. This setting allows testing unfavorable conditions for the robot that cannot be tested in real-world environments while ensuring human safety.

For each method and scenario configuration, 200 episodes were run. In the case of the two versions of SNGNN2D and GMM, the generated cost maps are used by a common control system in charge of computing the minimum cost path and moving the robot toward the goal position following that path.

Table 5 shows the mentioned metrics for each method across the two scenario configurations, with the best result for each metric highlighted in bold. Based on these results, SNGNN2D-v2 can be considered the proposal offering the most favorable social metrics. Thus, the presented approach provides the highest success rate, exhibiting a relatively smaller decline in this metric for the second configuration compared to the other methods. The main reason for a higher success rate is that the collision rate and, particularly, the human collision rate, is significantly lower in SNGNN2D-v2. In addition, in SNGNN2D-v1, GMM, and ORCA, most collisions with humans are caused by the robot.

ORCA is the method exhibiting the worst behavior regarding human collisions. This result could be expected since ORCA assumes that all agents have the same responsibility when it comes to avoiding collisions. Comparatively, SNGNN2D-v1 and GMM present similar success and collision rates for both configurations. These results underscore the limitations of both methods in considering the dynamics of the environment. In the case of SNGNN2D-v1, this limitation arises from using static data exclusively during model training. Besides, the model cannot represent dynamic interactions, causing the robot to traverse the interaction area between two humans walking together occasionally. Regarding GMM, besides the limitations in modelling certain aspects of the environment (such as the adaptability of the personal space according to people density), the frequency at which maps can be generated is dependent on the complexity

Table 5 Comparison between SNGNN2D-v2, SNGNN2D-v1, GMM and ORCA in simulated scenarios

	Configuration 1				Configuration 2			
	SNGNN2D-v2	SNGNN2D-v1	GMM	ORCA	SNGNN2D-v2	SNGNN2D-v1	GMM	ORCA
sr(%)	**91,00**	85,50	85,50	72,50	**87,00**	78,50	78,00	52,50
cr(%)	**7,00**	14,50	14,00	20,50	**9,50**	21,50	22,00	40,50
wcr(%)	2,00	0,50	**0,00**	5,50	2,00	1,50	**0,50**	7,00
ocr(%)	**0,00**	0,50	1,50	4,50	**0,50**	1,50	**0,50**	6,00
hcr(%)	**5,00**	13,50	12,50	10,50	**7,00**	18,50	21,00	27,50
rcp(%)	**40,00**	96,30	80,00	100,00	**50,00**	83,80	95,20	100,00
to(%)	2,00	**0,00**	0,50	7,00	3,50	**0,00**	**0,00**	7,00
fp(%)	4,20	2,50	**2,20**	14,10	4,50	**1,70**	2,50	19,60
st(s)	7,270	4,537	4,953	**0,000**	9,772	4,690	5,013	**0,444**
t(s)	16,776	14,464	14,959	**12,012**	20,121	14,259	15,403	**12,261**
pl(m)	3,939	3,383	**3,311**	4,191	4,421	3,265	**3,263**	4,278
spl(%)	**82,50**	82,20	81,90	70,10	**76,60**	**76,70**	74,60	50,00
psc(%)	**98,40**	97,70	98,00	98,10	**99,00**	97,60	97,00	95,20
dh_{min}(m)	**1,627**	1,375	1,546	1,507	**1,196**	1,150	1,157	1,066
cd_{min}(m)	1,254	**1,336**	1,307	1,104	1,166	1,263	**1,314**	1,178
cd_{avg}(m)	4,297	4,321	**4,273**	4,275	**4,340**	4,266	4,287	4,309
v_{min}(m/s)	0,000	0,000	0,000	0,232	0,000	0,000	0,000	0,215
v_{avg}(m/s)	0,230	0,241	0,224	0,315	0,214	0,234	0,217	0,299
v_{max}(m/s)	0,400	0,400	0,400	0,388	0,400	0,398	0,400	0,379

of the scenario. This restricts the robot's ability to replan in time to avoid collisions with moving humans. This limitation becomes evident when introducing more stationary humans to the environment configuration. Specifically, modifying the second configuration to include six additional static humans leads to a decline in GMM's success rate to 69.5%, mainly due to an increase in the human collision rate. However, the two versions of SNGNN2D exhibit comparable results to those of the original configuration, providing a success rate of 76.5% for SNGNN2D-v1 and 85.5% for SNGNN2D-v2.

Concerning navigation metrics, the time required to reach the goal and the path length are higher for SNGNN2D-v2 than for the other three methods (except the *pl* for configuration 1 when using ORCA). This is because the cost map in the second version of SNGNN2D forces a longer and more complex path –resulting in more directional changes– to minimize the discomfort of stationary humans and avoid collisions with dynamic humans. Consequently, metrics such as *spl*, dh_{min}, and *psc* indicate higher values for both configurations in SNGNN2D-v2, suggesting more socially acceptable robot behavior even in complex situations.

To complement these results with user opinions, we designed a survey featuring several videos of the robot navigating in simulated scenarios using GMM, SNGNN-v1, and SNGNN-v2. Participants were asked to rate these videos based on how well the robot navigated without disturbing the humans. The score ranged from 0 to 10, with 0 representing unacceptable robot behavior and 10 indicating perfect behavior.

The videos were randomly generated using SocNavGym, considering a $10m \times 10m$ room with various objects, a variable number of moving humans ranging from 2 to 6, and a variable number of stationary humans ranging from 0 to 6. To prevent *survey fatigue*, the survey was organized into three forms, each containing 15 videos-5 videos per method.

A total of 125 people participated in the survey. Specifically, 41 participants answered forms 1 and 3, while 43 answered form 2. Thus, each method received 625 scores. Among the participants, 60% were men and 40% were women, with ages ranging from 23 to 68 years. Although the majority had experience with modern technology, 76% had little to no experience with robots.

To analyse the inter-rater reliability, the Intraclass Correlation Coefficient (ICC) and its 95% confidence interval (CI95%) were calculated for each form individually. Particularly, we estimated ICC based on a mean-rating, absolute agreement, 2-way mixed-effects model (ICC3k) [30]. Table 6 summarizes these results, which show a high degree of agreement among the participants of each survey form.

The results of the ratings of each method are depicted in Fig. 9 and Table 7. Figure 9 presents a box plot showing the distribution of the scores for the three methods, while Table 7

Table 6 Results of the Intraclass Correlation Coefficient for the three survey forms

Form	ICC3k	CI95%
1	0.995393	[0.99, 1.0]
2	0.993564	[0.99, 1.0]
3	0.991241	[0.98, 1.0]

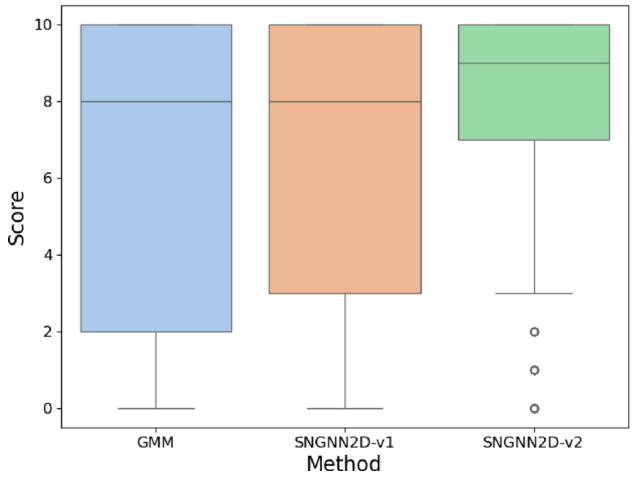


Fig. 9 Box plot showing the distribution of scores per method

Table 7 Descriptive statistics (mean, median, and standard deviation) of the survey scores of GMM, SNGNN2D-v1, and SNGNN2D-v2

Method	Mean	Median	SD
GMM	6.2896	8.0	4.045134
SNGNN2D-v1	6.7040	8.0	3.726688
SNGNN2D-v2	7.9168	9.0	2.911536

provides their descriptive statistics. Both illustrations indicate SNGNN2D-v2 received higher ratings compared to the other two methods, which can be considered comparable. To determine the significance of these results, we conducted an ANOVA test, which yielded a p-value of less than 0.001, followed by a post-hoc Tukey's HSD test (Table 8). The results of the Tukey test show that the difference between GMM and SNGNN2D-v1 is not significant, suggesting that the survey participants found these two methods to be roughly equivalent. However, both GMM and SNGNN2D-v1 were significantly outperformed by SNGNN2D-v2, which was perceived as causing less discomfort to humans according to the evaluations.

4.4 Comparison with Pure CNN Based Model

The last experiment presents the results of cost map generation utilising a CNN-based model, in contrast with the proposed model. As mentioned, the model selected for comparative evaluation is *Pix2pix* ([23]), a remarkable Generative Adversarial Network (GAN) capable of transposing an image into another. As the proposed model, *Pix2pix* was trained utilising an identical dataset, achieving a Mean Squared Error (MSE) of **0.0084926** within the test set, which is roughly double the MSE produced by SNGNN2D-v2 in the same set.

For *Pix2pix*, the input volumes are constructed by concatenating three video frames from the dataset along the channel dimension. These frames have a temporal difference of one second, mirroring the method used to construct the graph for the SNGNN2D-v2.

Figure 10 provides a visual comparison of the *Pix2pix* results alongside the corresponding ground truth used for training, including a frame from the source video on the left-hand side. The output generated by SNGNN2D-v2 has also been included on the right-hand side of Fig. 10. As can be observed, the outcomes bear high visual similarity and closely align with the results generated by the proposed model. Nevertheless, the SNGNN2D-v2 model asserts two primary advantages over CNN-based models: superior representation of entity interactions and invariance to changes in the appearance of the environment and its elements. These benefits are demonstrated across two different experiments.

In the initial experiment, *Pix2pix* is tasked with generating cost maps for two scenarios from the test set in which two individuals are interacting. The resultant maps can be found in Fig. 11, where *Pix2pix* incorrectly models interaction areas where no interaction exists and vice-versa. In fact, this incorrect modeling of interaction zones is the primary cause of the difference in the MSE for the test set between both approaches. Extending the dataset with more interaction samples would help improve the model. However, this approach would limit interaction detection to visual cues alone. Relationships between entities could also be identified using other perceptual sources. For instance, hearing two people talking might help infer that they are interacting.

In the second experiment, the texture and color of the room floor of the scenarios were altered, and 50 new samples of this type of scenario were generated. Figure 12 displays the outcomes for two distinct examples. In this case, the output image produced by *Pix2pix* is nearly unidentifiable due to its heavy reliance on visual features. In contrast, the output generated by SNGNN2D-v2 remains unaffected by the transformation in the floor pattern. This result is also confirmed quantitatively through the MSE between the output generated by the two models and the ground truth. While the MSE of SNGNN2D-v2 keeps a very low value (0.002759), the MSE of *Pix2pix* increases significantly to 0.1262. Similar to *Pix2pix*, SNGNN2D-v2 requires detecting objects and humans with varying appearances. However, for accurate map generation using only visual information, the combination of different appearance elements to form a representative set of potential scenarios would need to be extensive and visually realistic. Leveraging individual detectors for identifying

Table 8 Results of the Tukey's HSD test for the survey data

Group 1	Group 2	Mean difference	p-adjusted	CI95%
GMM	SNGNN2D-v1	0.4144	0.1033	[−0.0623, 0.8911]
GMM	SNGNN2D-v2	1.6272	< 0.001	[1.1505, 2.1039]
SNGNN2D-v1	SNGNN2D-v2	1.2128	< 0.001	[0.7361, 1.6895]

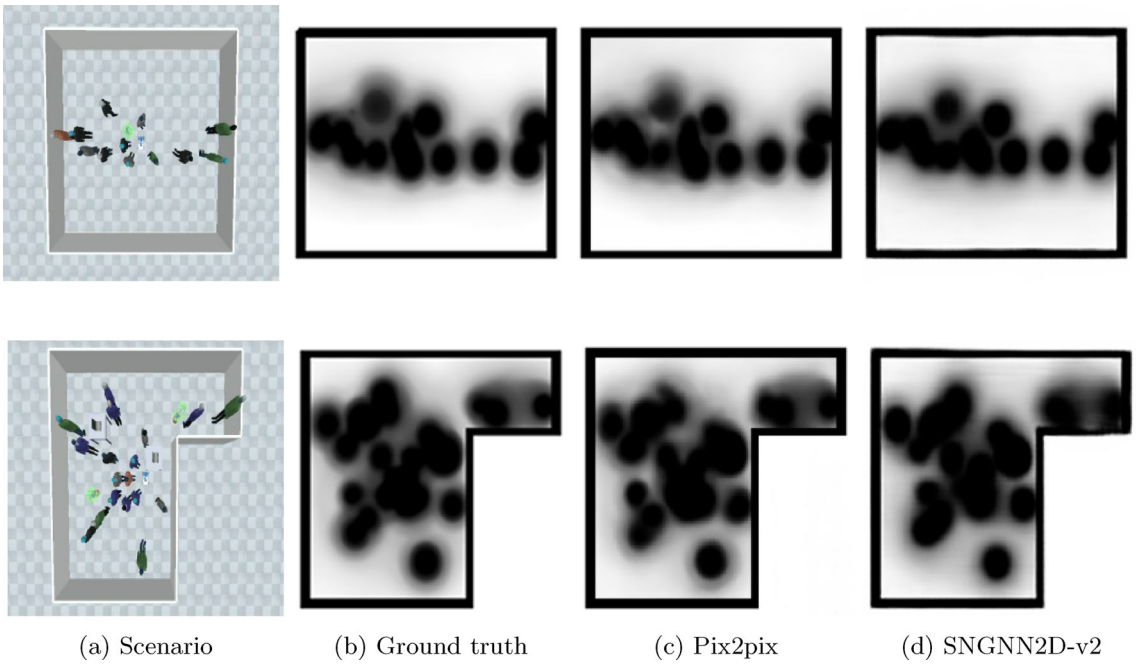


Fig. 10 Disruption maps generated by Pix2pix (c) and SNGNN2D-v2 (d) for the scenarios in (a). The ground truth is shown in (b)

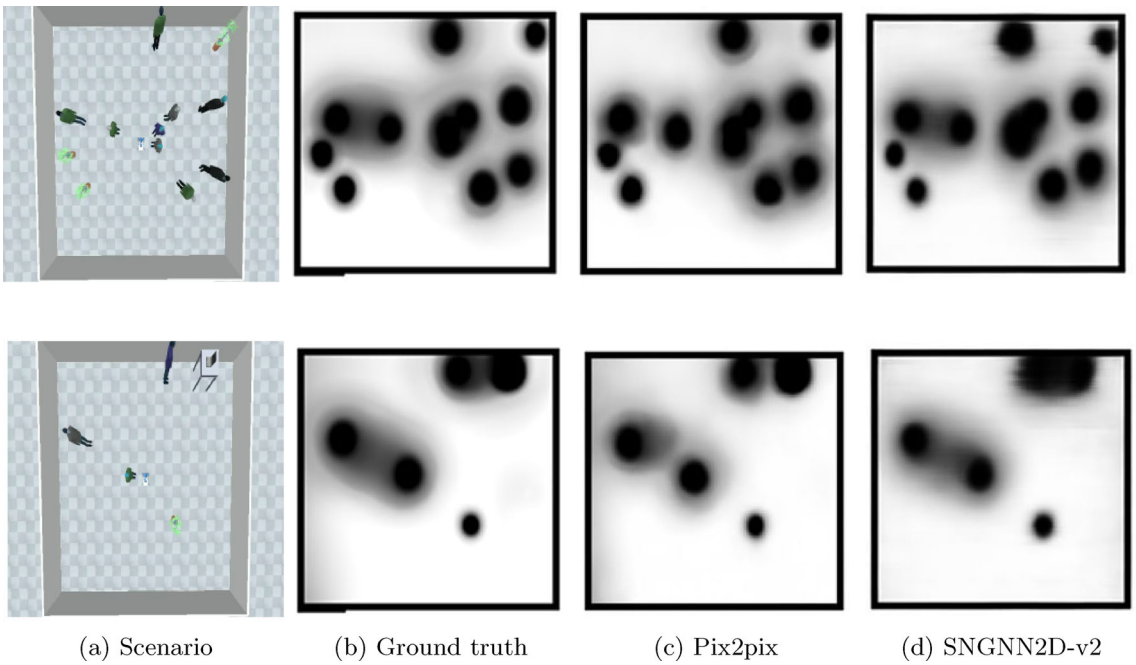


Fig. 11 Comparison between the cost maps generated by Pix2pix and SNGNN2D-v2 in scenarios with interactions

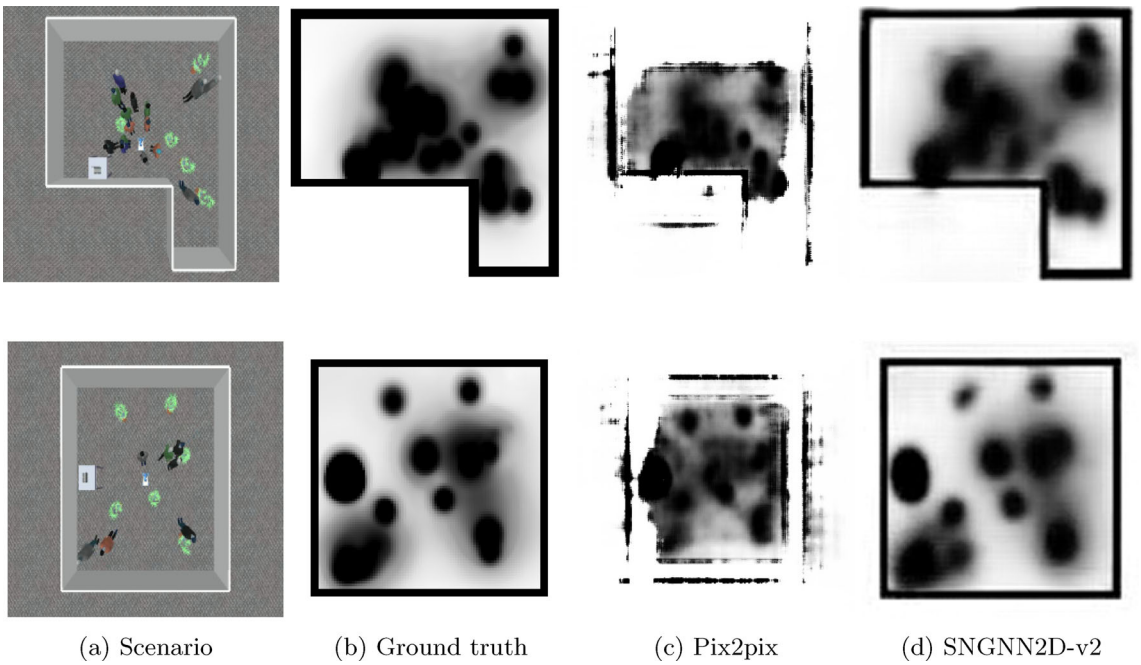


Fig. 12 Comparison between the cost maps generated by Pix2pix and SNGNN2D-v2 in scenarios with a different floor than the scenarios used during training

entities in the environment allows for the use of a significantly smaller dataset to build a functional model.

5 Conclusion

Navigation cost maps are a critical component of human-aware navigation (HAN) systems, providing robots with valuable information for navigating shared environments while minimizing social disruption. However, manually crafting cost maps accurately reflecting dynamic environments and human interactions is complex and time-consuming. To address this challenge, we propose a learning-based approach using graph neural networks (GNNs) to generate cost maps in real time. Our model, SNGNN2D-v2, takes entities' positions, velocities, and interactions in the environment as input and effectively encodes social disturbance areas and interaction zones.

The model can be retrained using the same strategy to account for new situations not explicitly included in the current training set, such as larger objects or navigation areas. Additionally, the generated cost maps can be combined with other types of maps (e.g., occupancy grids) to adjust the final map to the specific environment.

Experimentation indicates that incorporating entity velocities into the model significantly enhances its performance compared to its predecessor (SNGNN2D-v1 [9]) and non-learning-based approaches. Additionally, the comparison with a convolutional neural network (CNN)-based approach [23] trained on the same dataset indicates potential advantages of GNNs. The use of structured data instead of raw images allows the model to remain invariant to changes in scenario appearance and to model interaction areas effectively with a relatively moderate-sized dataset. Regarding real-world experiments, the proposed model shows the ability to generate maps in real time, enabling it to respond appropriately to the dynamics of the environment. The results also suggest the model can adapt conveniently to situations where the available space is limited. However, the number and complexity of scenarios need to be extended to confirm these findings and complete the validation of the model for its use in real environments.

Our future work aims to extend the dataset to improve the generalizability of the model by incorporating the following additional data: real-world samples with noisy detections and more realistic human motions; a wider variety of obstacles and navigation areas; and additional human-centered information, such as gaze direction and ongoing activities. Furthermore, we aim to improve the functionality of our model by adapting its output to generate different cost maps corresponding to various robot actions. This refinement will allow the creation of paths with associated robot speeds at each step, which will enhance the robot's social adaptability and responsiveness in dynamic environments.

Supplementary information

The data and models that support the findings of this paper have been made publicly available at https://www.dropbox.com/scl/fo/k282y10fecljyyl7sjj10/h?rlkey=e1i96zi1nqpfb50k2xh5aq9tx&dl=0. The code is available in a public GitHub repository at https://github.com/gnns4hri/SNGNN2D-v2.

Acknowledgements Experiments were run on Aston EPS Machine Learning Server, funded by the EPSRC Core Equipment Fund, Grant EP/V036106/1. This work was also supported by the Spanish Government under Grants PID2022-137344OB-C31, TED2021-131739B-C22 and PDC2022-133597-C41.

Author Contributions DRC, PBB and LJM were mainly involved in conceptualization, formal analysis and software development. Supervision was provided by PBB and LJM, while validation was conducted by DRC and LVC. The original draft of the manuscript was prepared by DRC, with subsequent review and editing by LVC, PBB and LJM. All authors read and approved the final version of the manuscript.

Funding Experiments were run on Aston EPS Machine Learning Server, funded by the EPSRC Core Equipment Fund, Grant EP/V036106/1. This work was also supported by the Spanish Government under Grants PID2022-137344OB-C31, TED2021-131739B-C22 and PDC2022-133597-C41. Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data Availability https://www.dropbox.com/scl/fo/k282y10fecljyyl7sjj10/h?rlkey=e1i96zi1nqpfb50k2xh5aq9tx&dl=0

Code Availability https://github.com/gnns4hri/SNGNN2D-v2

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Gross HM, Scheidig A, Müller S, Schütz B, Fricke C, Meyer S (2019) Living with a Mobile Companion Robot in your Own Apartment - Final Implementation and Results of a 20-Weeks Field Study with 20 Seniors. In: 2019 international conference on robotics and automation (ICRA); pp 2253–2259
2. Chen TL, Ciocarlie M, Cousins S, Grice P, Hawkins K, Hsiao K et al (2013) Robots for humanity: using assistive robotics to empower people with disabilities. IEEE Robotics Autom Mag 20(1):30–39. https://doi.org/10.1109/MRA.2012.2229950
3. Bradwell HL, Edwards KJ, Winnington R, Thill S, Jones RB (2019) Companion robots for older people: importance of user-centred design demonstrated through observations and focus groups comparing preferences of older people and roboticists in South West England. BMJ Open. 9(9):e032468. https://doi.org/10.1136/bmjopen-2019-032468
4. Singamaneni PT, Bachiller-Burgos P, Manso LJ, Garrell A, Sanfeliu A, Spalanzani A et al (2024) A survey on socially aware robot navigation: taxonomy and future challenges. Int J Robot Res 43:02783649241230562. https://doi.org/10.1177/02783649241230562
5. Kruse T, Pandey AK, Alami R, Kirsch A (2013) Human-aware robot navigation: a survey. Robot Auton Syst 61(12):1726–1743. https://doi.org/10.1016/j.robot.2013.05.007
6. Möller R, Furnari A, Battiato S, Härmä A, Farinella GM (2021) A survey on human-aware robot navigation. Robot Auton Syst 145:103837. https://doi.org/10.1016/j.robot.2021.103837
7. Manso LJ, Jorvekar RR, Faria DR, Bustos P (2021) Graph Neural Networks for Human-Aware Social Navigation. In: Advances in Physical Agents II. Springer International Publishing. Cham pp. 167–179
8. Manso LJ, Nuñez P, Calderita LV, Faria DR, Bachiller P (2020) SocNav1: a dataset to benchmark and learn social navigation conventions. Data 5(1):7. https://doi.org/10.3390/data5010007
9. Rodriguez-Criado D, Bachiller P, Manso LJ (2021) Generation of human-aware navigation maps using graph neural networks. In: international conference on innovative techniques and applications of artificial intelligence. Springer, Cham; pp. 19–32
10. Bachiller P, Rodriguez-Criado D, Jorvekar RR, Bustos P, Faria DR, Manso LJ (2022) A graph neural network to model disruption in human-aware robot navigation. Multimed Tools Appl. https://doi.org/10.1007/s11042-021-11113-6
11. Vega A, Manso LJ, Macharet DG, Bustos P, Núñez P (2019) Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. Pattern Recognit Lett. 118:72–84. https://doi.org/10.1016/j.patrec.2018.07.015
12. van den Berg J, Guy SJ, Lin M, Manocha D (2011) Reciprocal n-body collision avoidance. In: Pradalier C, Siegwart R, Hirzinger G (eds) Robot Res. Berlin, Heidelberg, Springer, Berlin Heidelberg, pp 3–19
13. Charalampous K, Kostavelis I, Gasteratos A (2017) Recent trends in social aware robot navigation: a survey. Robot Auton Syst 93:85–104. https://doi.org/10.1016/j.robot.2017.03.002
14. Dondrup C, Hanheide M (2016) Qualitative constraints for human-aware robot navigation using velocity costmaps. In, (2016) 25th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE 586–592
15. 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); p. 1–6
16. Laible S, Zell A (2014) Building local terrain maps using spatio-temporal classification for semantic robot localization. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; p. 4591–4597
17. 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. https://doi.org/10.1007/s41315-020-00125-4

18. Li J, Zhang F, Fu Y (2024) A novel Human-Aware Navigation algorithm based on behavioral intention cognition. Robotica. https://doi.org/10.1017/S0263574723001832
19. Hemachandra S, Walter MR, Tellex S, Teller S (2014) Learning spatial-semantic representations from natural language descriptions and scene classifications. In: 2014 IEEE international conference on robotics and automation (ICRA); pp 2623–2630
20. Kostavelis I, Charalampous K, Gasteratos A, Tsotsos JK (2016) Robot navigation via spatial and temporal coherent semantic maps. Eng Appl Artif Intell 48:173–187. https://doi.org/10.1016/j.engappai.2015.11.004
21. Vasquez D, Okal B, Arras KO (2014) Inverse Reinforcement Learning algorithms and features for robot navigation in crowds: An experimental comparison. In: 2014 IEEE/RSJ international conference on intelligent robots and systems; pp 1341–1346
22. Shiying Sun QL, Zhao Xiaoguang, Tan M (2020) Inverse reinforcement learning-based time-dependent A* planner for human-aware robot navigation with local vision. Adv Robot 34(13):888–901. https://doi.org/10.1080/01691864.2020.1753569
23. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-Image Translation with Conditional Adversarial Networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR); pp 5967–5976
24. Baghel R, Kapoor A, Bachiller P, Jorvekar RR, Rodriguez-Criado D, Manso LJ (2020) A toolkit to generate social navigation datasets. Workshop Phys Agents. Springer, Newyork, pp 180–193
25. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al (2018) Relational inductive biases, deep learning, and graph networks. arXiv: 1806.01261. pp 1–40. doi 10.1017/S0031182005008516
26. Kapoor A, Swamy S, Bachiller P, Manso LJ (2023) SocNavGym: A Reinforcement Learning Gym for Social Navigation. In: 32nd IEEE international conference on robot and human interactive communication, RO-MAN 2023, Busan, Republic of Korea, 28-31, IEEE; pp 2010–2017
27. Rösmann C, Feiten W, Wösch T, Hoffmann F, Bertram T (2012) Trajectory modification considering dynamic constraints of autonomous robots. In: ROBOTIK 2012; 7th German conference on robotics. VDE; p. 1–6. Available from: https://ieeexplore.ieee.org/document/6309484
28. Rodriguez-Criado D, Bachiller P, Vogiatzis G, Manso LJ (2024) Multi-person 3D pose estimation from unlabelled data. Mach Vision Appl. https://doi.org/10.1007/978-3-030-01240-3_25
29. Francis A, Pérez-D'Arpino C, Li C, Xia F, Alahi A, Alami R, et al. Principles and Guidelines for Evaluating Social Robot Navigation Algorithms
30. Koo TK, Li MY (2016) A guideline of selecting and reporting intraclass correlation coefficients for reliability research. J chiropr med 15(2):155–163

Daniel Rodriguez-Criado received his B.S. and M.S. degrees in industrial engineering from the University of Malaga, Malaga, Spain, in 2015 and 2017, respectively. Additionally, in 2019, he received an M.S. degree in electronic systems for sensorised environments jointly at the University of Malaga and Hong Kong University of Science and Technology (HKUST) in Hong Kong. He studied for his Ph.D. in Computer Science at Aston University, Birmingham, United Kingdom, receiving the title in 2024. His research interest spans three areas where the main topic is the applications of Graph Neural Networks: Human-aware navigation with robots, 3D human pose estimation with RGB cameras and image generation for intelligent transportation systems.

Pilar Bachiller-Burgos is an Associate Professor at the University of Extremadura (UEx), Spain. She received her B.S., M.S., and Ph.D. degrees in Computer Science from the UEx in 1997, 2000 and 2008, respectively. She has been a member of the RoboLab research group since its inception and has been its leader since 2021. Within RoboLab, she has contributed to numerous projects focused on the design and programming of robots for social, educational and assistive purposes. Her research interests include robotics, computer vision, active perception, software engineering for robotics, and machine learning.

Luis V. Calderita is an Assistant Professor at the Department of Computer and Telematic Systems Engineering, University of Extremadura. He earned his B.S. in Computer Science in 2009 and Ph.D. in 2016, both from the University of Extremadura, and an M.S. in Intelligent Systems from the University of Salamanca in 2010. He has also served as an Assistant Professor at the University of Málaga and the University of León. His research focuses on social robotics, human-robot interaction, and robotics software engineering. He has authored 43 publications, including 16 in international journals

Luis J. Manso is a Senior Lecturer in Computer Science at the Applied Artificial Intelligence and Robotics Department, Aston University, and part of the Autonomous Robotics and Perception research group. He obtained his Computer Engineering degree in 2009 and his PhD in 2013, both from the University of Extremadura. His research interests include geometric learning, active perception, human-robot interaction and sparse predictive world models.