Received November 23, 2021, accepted December 19, 2021, date of publication December 24, 2021,
date of current version January 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3138614

Efficient Prediction of Human Motion for Real-Time Robotics Applications With Physics-Inspired Neural Networks

ALESSANDRO ANTONUCCI[1], GASTONE PIETRO ROSATI PAPINI[2], PAOLO BEVILACQUA[1], LUIGI PALOPOLI[1], (Member, IEEE), AND DANIELE FONTANELLI[2], (Senior Member, IEEE)

[1]Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy
[2]Department of Industrial Engineering, University of Trento, 38123 Trento, Italy

Corresponding author: Alessandro Antonucci (alessandro.antonucci@unitn.it)

ABSTRACT Generating accurate and efficient predictions for the motion of the humans present in the scene is key to the development of effective motion planning algorithms for robots moving in promiscuous areas, where wrong planning decisions could generate safety hazard or simply make the presence of the robot ‘‘socially’’ unacceptable. Our approach to predict human motion is based on a neural network of a peculiar kind. Contrary to conventional deep neural networks, our network embeds in its structure the popular Social Force Model, a dynamic equation describing the motion in physical terms. This choice allows us to concentrate the learning phase in the aspects which are really unknown (i.e., the model's parameters) and to keep the structure of the network simple and manageable. As a result, we are able to obtain a good prediction accuracy even by using a small and synthetically generated training set. Importantly, the prediction accuracy remains acceptable even when the network is applied in scenarios radically different from those for which it was trained. Finally, the choices of the network are ‘‘explainable’’, as they can be interpreted in physical terms. Comparative and experimental results prove the effectiveness of the proposed approach.

INDEX TERMS Human motion prediction, neural networks, social force model, service robotics.

I. INTRODUCTION

We are creating a new generation of robots that can move freely, take autonomous decisions and interact directly with humans to execute tasks and deliver services. The level of the expectations raised by this development is very high. People imagine a near future in which robots and humans will share public spaces, talk to each other, exchange objects and, to some extent, create emotional links. Is this a realistic perspective? In order to be up to the task, robots are required to be efficient for the tasks they are assigned (why else should we use a robot?) and are required to be safe. The vocal reaction to the relatively small number of accidents caused by autonomous cars in the past years shows that for robots the bar of safety requirements has to be set very high. However, in order to move between humans, safety is not enough: a robot has to be smooth in its reactions, it has to respect the personal space of the humans as much as possible, and it has to be predictable. Humans have sophisticated means to move in a socially acceptable way. Their social intelligence allows them to predict the intent of their peers decoding a complex non spoken language made of gestures, facial expressions, gaze, pose of the limbs.

Creating algorithms that replicate a part of this broad set of abilities compounding social intelligence is the holy grail of human-robot interaction and the topic of this paper. Specifically, our objective is to *predict where a person intends to go by simply looking at how s/he moved in the immediate past and at the shape of the environment*. Our overarching goal is to use these predictions in order to create human-aware motion planning algorithm, a research area that is becoming increasingly popular [1], [2]. Our specific take is to use accurate predictions of human motions for robot plan synthesis in order to be inherently safe and compliant with unwritten social rules.

A. RELATED WORK

1) PHYSICS BASED APPROACHES

Many physics based models have been proposed to predict human motion in a social context, e.g. [3]. The most famous is the Social Force Model (SFM) [4]. In the SFM a person

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro Neto.

is seen as a particle acted on by attractive forces (the goals) and repulsive forces (the obstacles). Since it is simple to implement and yet highly effective in the representation of the individual motion of human beings, it has been adopted by a number of research works. One peculiar feature is that the attractive and repulsive forces of the SFM can be generalised by geometric information only, while the resulting motion effect is a linear combination of these elements. The model however has known limitations. One of the most important is that modelling a person as a particle does not differentiate between motion patterns that are "natural" and others that are possible but not frequently taken (e.g., sideways motions). These issues can be addressed by leveraging a relatively high sampling rate and/or by integrating the preferential nonholonomic behaviour of the human motion into the model [5], [6].

An important problem to tackle in order to use the SFM is how to estimate its many parameters and in particular the intensity and the direction of the attractive and of the repulsive forces that animate the motion. A first possibility is to make heuristic "rule-of-the-thumb" choices, but this option is workable only in very specific conditions, e.g., interaction between a robot and a human in free space [7], [8]. Another issue is how to estimate the target of walking pedestrian from the past motion [9]. For example, in [10], a virtual goal is chosen as the position that a person would reach if s/he moved with constant velocity, while in [11] a set of trajectory sub-goals are estimated from the recorded data in a structured environment. Furthermore, [12] proposed a modified formulation of the SFM to calibrate the parameters with observable features from empirical data.

2) NEURAL NETWORKS APPROACHES

Neural Networks (NN) hold the promise to master the complexity of predicting the intention of humans in a relatively simple way. Similarly to our work, [13] modelled an area of visual attention and social interaction for the pedestrian, jointed to the head orientation, and used to strengthen the training of a Long Short Term Memory network (LSTM). A deep learning-based classifier is used in [14] to learn behaviour patterns from visual cues is mixed with a game theory model encoding the SFM to forecast the interaction between multiple pedestrians. A common approach to manage multiple future trajectories has been to generate different motion modes. For this reason, newly learning approaches implement multiple predictions to describe mixed motion behaviours. In [15] a Generative Adversarial Network (GAN) based approach is exploited with a novel social pooling framework to predict multiple trajectories while learning social norms. A similar GAN based framework with a social attention mechanism is proposed by [16]. Both these approaches utilize pedestrians past trajectories and scene context information, but do not consider the agents' destinations. Conversely, a recent work by [17] uses Variational Autoencoder (VAE) based network to infer a distribution of waypoints and obtain a multi-modal trajectory prediction, while [18] reformulated maximum entropy IRL to jointly infer waypoints and human trajectories on a 2D grid defined over the scene.

In principle, a deep neural network (DNN) trained with a sufficient number of samples could learn the human motion patterns by discovering the underlying dynamic model on its own. However, the number of layers and of neurons required to manage the complexity of human behaviours can be very large and is anyway hard to predict. Equally difficult is to understand the number of samples that are needed to train a network of this complexity. Finally, the use of a DNN lacks a property of remarkable importance for many applications: the so-called "explainability". When an autonomous system takes a decision it is important to understand why that specific choice has been made, in order to solve bugs or attribute legal responsibilities [19]. The total absence of a prior model in a DNN makes explainability hard or even impossible to achieve. Another known limitation shared by NN approaches is their difficult training. The available annotated data sets are not many, and the parameters overfitting is an actual risk when data have strong similarities. As a result, as shown by [20], neural models can easily be outperformed by a simple constant velocity model in the case of linear trajectories.

3) PHYSICS AND NEURAL NETWORK MIXED APPROACHES

Past attempts to use machine learning techniques in combination with physical models have applied gradient descend methods to learn the interaction forces [21], used linear regression and NN to predict the direction of motion [22], used evolutionary algorithm to optimise the SFM parameters from video segments [23]. The approach taken in this paper is rather new and is inspired to the simulation of vehicle dynamics by [24].

4) GOALS SELECTION

The motion of a person is driven by her intent, i.e., by where she intends to go or perform an action. Dynamic goal inference based on the semantic of the environment is still an open issue [9]. Most existing works rely on a predefined set of goals, estimated from observed trajectory data. For example, goals can be deducted from the prevailing direction taken by the pedestrians, by noting the preferred locations where they stop, or by partitioning the environment with a Voronoi-based method [11], [25]. On the other hand, a few papers have sought to identify the goals from real-time motion predictions. For instance, [26] proposed a heuristic method to automatically determine goal positions on a 2D semantic grid map. Cell transitions are predicted through discrete Markov chains. In [27], Voronoi partitions [11] were used in order to obtain a good guess of the preferred sub-goal within a shopping mall. Their algorithm estimates the candidate goal by weighing the visibility and reachability of the sub-goals, and by using a service robot that moves side-by-side with the person. A recurrent Mixture Density Network (RMDN) was proposed by [28] to learn a mixture of potential destinations, which is fed into a Fully Convolutional Neural Network (CNN) to predict the human trajectories. In [29] they applied probabilistic Directional Grid Maps (DGM) in order to fit a mixture of von-Mises distributions of motion directions attributed to each grid cell of a discretised environment.

B. PAPER CONTRIBUTION

In this paper, we seek to bridge the gap between model based and learning based approaches in order to retain the advantages of both. Our goal is to predict the motion of a human for several seconds ahead using a short segment of past observations. The philosophy that underlies our approach is that learning is a powerful tool, but, when applied to human motion prediction, it should be used with some grain of salt. Indeed, the complexity of the task requires massive amounts of training data and the risk of overfitting the models is very concrete. With these considerations in mind, our approach restricts the application of learning to the sub-problems that are very difficult to tackle otherwise. In this class falls the estimation of the parameters of the SFM. Our idea is to use a NN structured so that its connections reflect the dynamics of the SFM. In other words, we embed our prior knowledge into the NN in the form of a model assuming that the latter, by a correct choice of parameters, closely approximates the dynamics of human motion. This way, the learning phase is concentrated on the aspects for which we actually lack any real knowledge: the SFM parameters and the virtual forces acting in the SFM.

The selection of the goals of the human motion, which is a key input for the SFM, deserves a special attention. As apparent from the survey reported above, this problem is way simpler than the estimation of the SFM parameters and force, and established techniques exist that are easy to implement and that can provide useful probabilistic information on the potential goals. In our work, we consider the following scenarios. The trivial case is when the possible goals are given or annotated on the map based on the analysis of the places of interest. On the opposite side of the spectrum, we consider a situation in which no prior information on the environment is available. In this case, we infer the position of the possible goals from the geometry of the activity space [30], from the motion of the humans and from the configuration of the static obstacles. Since for all of these scenarios, we can have multiple potential goals, we use a multi-goal approach based on the generation of different hypotheses and on the evaluation of the likelihood of the trajectories associated with each of them.

The advantages of the approach proposed in the paper are manifold: 1. wiring a model inside the NN reduces the number of neurons by a significant amount (we estimate one or two orders of magnitude), 2. using maximum likelihood evaluation of multiple hypotheses for the final goal of the target is a natural complement of the idea: the complexity is much lower than a monolithic NN solution, the training is simplified and the probabilities associated with each possible destination can be used in *what-if* motion planning solutions; 3. as shown in our experiments, a relatively small number of synthetically generated samples is sufficient to generate accurate predictions, even for scenarios that are quite different from the ones considered in the training set, thus the solution is very practical for robotic applications that involve navigation across indoor unknown scenarios; 4. because our NN retains the model inside, its decisions can be explained in physical terms, which simplifies the interpretation of the results of the NN and the explanation of its possible errors.

The paper is organised as follows. In Section II, we summarise some background knowledge on the SFM, which will prove useful in the development of the paper. In Section III, we report the key contribution of the paper: how to embed the SFM into the structure of a NN. In Section IV, we describe the multi-goal approach that completes our motion prediction framework. In Section V, we report a full set of experiments proving the validity of the approach. In Section VI, we describe the robotic platform employed for real-world experiments and present the corresponding results. Finally, in Section VII we state our conclusions and announce future work directions.

II. PRELIMINARIES

In this section, we briefly review necessary background material on the Social Force Model. In the SFM [4], the i-th pedestrian is modelled as a particle with mass m_i and radius r_i. Its position is denoted as $\mathbf{p}_i = [x_i, y_i]^T$ and is expressed in the frame $\langle F \rangle = \{X_f, Y_f\}$. The human moves towards his/her target at a certain desired walking speed with magnitude v_i^d and following a second order dynamic. At the same time, the motion is perturbed by the environment, e.g. fixed obstacles, walls, furniture, etc., and other agents in the environment. Omitting the subscript i for readability, the total force $\mathbf{f}$ that acts on the i-th pedestrian is given by $\mathbf{f} = \mathbf{f}^o + \mathbf{f}^e$, i.e.

$$m\dot{\mathbf{v}} = \mathbf{f}^o + \sum\nolimits_{j(\neq i)} \mathbf{f}_j^p + \sum\nolimits_k \mathbf{f}_k^w, \tag{1}$$

where $\mathbf{v} = \dot{\mathbf{p}}$. Moreover, the attractive force $\mathbf{f}^o$ is defined as

$$\mathbf{f}^o = (m/\tau)\left[v^d(t)\mathbf{e}^d(t) - \mathbf{v}(t)\right] \tag{2}$$

where the characteristic time $\tau > 0$ parameter determines the rate of change of the velocity vector, while $\mathbf{e}^d$ is the unit vector pointing towards the goal. The force exerted by the static obstacle k on the i-th pedestrian is given by

$$\mathbf{f}_k^w = Ae^{(r-d_w)/B}\mathbf{n}_k + k_1 g\,(r - d_k)\,\mathbf{n}_k + \\ - k_2 g\,(r - d_k)\,(\mathbf{v} \cdot \mathbf{t}_k)\,\mathbf{t}_k, \tag{3}$$

i.e. it is the sum of a repulsive component, a compression force and a sliding friction force. We denote by $d_k = ||\mathbf{p} - \mathbf{p}_k||$ the distance between the pedestrian centre of mass and the coordinates of the obstacle closest point, so that $\mathbf{n}_w = (\mathbf{p} - \mathbf{p}_k)/d_k$ and $\mathbf{t}_k = [-\mathbf{n}_k(2), \mathbf{n}_k(1)]^T$ are the distance unit vector and its tangential direction, respectively. The function $g(x) = \max\{0, x\}$ models the fact that both the compression and the sliding friction forces exist only if the pedestrian touches the obstacle (i.e., $d_k > r$). A, B, k_1 and k_2 are the model parameters. Notice that in this paper we are neglecting the interaction forces $\mathbf{f}_j^p$ with the j-th pedestrian in (1), which will be the objective of future work.

III. HUMAN MOTION PREDICTION

Human motion predictions are generated using an embedding of the SFM into a structured neural network, i.e., a NN organised such that the neurons process the input signals according to (1). Two separate branches are designed to estimate the SFM forces in two different scenarios. In the case of the *open environment* scenario, the agent moves freely towards its goal, so it is subject only to the force term in (2). In the second scenario, the pedestrian moves across a space cluttered with obstacles and is affected by the repulsive force (3). We will henceforth refer to this scenario as *structured environment*. Consequently, each network branch models effects of different nature, i.e. attractive or repulsive forces, which are summed up at the end to produce the resulting force.

A. OPEN ENVIRONMENT

While freely moving towards the desired goal, the pedestrian is only affected by the force $\mathbf{f}^o$ in (2). Hence, the first branch of the neural network (named *Net1*) is used to predict the two force components f_x^o, f_y^o. The network inputs are the n more recent samples of the past $\mathbf{p}$ coordinates of the pedestrian (from the current time t) obtained with a sampling period δ_t. In order to avoid spatial biases, the coordinates are shifted with respect to the first sample of the window, so as to generate the input vector $\Delta\mathbf{p}(t) \in \mathbb{R}^{2\times n}$ as in the following equation

$$\begin{aligned}\Delta\mathbf{p}(t) = [\mathbf{p}(t-(n-1)\delta_t), \mathbf{p}(t-(n-2)\delta_t), \ldots, \mathbf{p}(t)] +& \\ -\mathbf{p}(t-(n-1)\delta_t)1_n,& \quad (4)\end{aligned}$$

where 1_n is an n-dimensional column vector with all ones. The first part of *Net1* consists of two hidden layers with no biases and with only one fully connected output neuron that learns the instantaneous velocity v_x, v_y on the X_f and Y_f axis, respectively. These two layers are followed by a tanh$(\cdot)$ activation function. Another neuron with no bias follows each layer in order to facilitate the convergence of the estimates to their actual range. Moreover, the most recent relative motion measurement $\Delta^n\mathbf{p}(t) = \mathbf{p}(t) - \mathbf{p}(t-1)$ is used to estimate the components of the normalised unit vector $\mathbf{e}^d$ pointing to the goal. Finally, the vector of the desired velocity v^d is derived from the velocity magnitudes $\mathcal{V}(t) \in \mathbb{R}^n$ estimated by the following

$$\begin{aligned}\Delta\mathbf{p}'(t) &= \left[\Delta^2\mathbf{p}, \ldots, \Delta^n\mathbf{p}\right] - \left[\Delta^1\mathbf{p}, \ldots, \Delta^{n-1}\mathbf{p}\right], \\ \mathcal{V}(t) &= \left[||\Delta^1\mathbf{p}'(t)||, \ldots, ||\Delta^{n-1}\mathbf{p}'(t)||\right]. \qquad (5)\end{aligned}$$

All the estimates pass through a Lambda layer where they are combined and weighted according to the m and τ parameters in (2). The *Net1* output is then the estimate of $\mathbf{f}^o = \left[f_x^o, f_y^o\right]^T$. The force input (2) is translated in the form of a structured NN by the following

$$\mathbf{f}^o = \overbrace{\text{sig}\left(\mathcal{V}(t)W_v\right) w_{vs}}^{\frac{mv^d}{\tau}} \overbrace{\frac{\Delta^n\mathbf{p}(t)}{||\Delta^n\mathbf{p}(t)||}}^{\mathbf{e}^d(t)} - \overbrace{\tanh\left(\Delta\mathbf{p}(t)W_{\mathbf{v}}\right) \odot w_{\mathbf{v}s}}^{\frac{\mathbf{v}(t)m}{\tau}}, \tag{6}$$

where $W_v \in \mathbb{R}^{10\times n-1}$, $w_{vs} \in \mathbb{R}^{10}$, $W_{\mathbf{v}} \in \mathbb{R}^{2\times 20}$, $w_{\mathbf{v}s} \in \mathbb{R}^{1\times 2}$, are the weight matrices and $\odot$ is the Hadamard product. The sig$(\cdot)$ sigmoid activation function is used to keep a positive sign for the v^d estimate, which then are rescaled by the weight matrix w_{vs}. The number of learnable parameters for *Net1* (represented in Fig. 1-a) is 123, where 100 of them are entirely devoted to the desired velocity estimates.

B. STRUCTURED ENVIRONMENT

For the environment with obstacles, the second branch *Net2* has two parallel sub-branches to predict $\mathbf{f}^o = \left[f_x^o, f_y^o\right]^T$ (described above) and $\mathbf{f}_k^w = \left[f_x^w, f_y^w\right]^T$ (reported in (3)) components of the force, respectively. While in the case of open environment the direction of the motion offers a good clue on the goal of the pedestrian, this is not the case in presence of obstacles. In fact, a given direction of motion can be chosen either because it leads to the goal or because it is a good way to avoid the obstacles [31]. For this reason, we directly provide $\mathbf{e}^d$ as input to the sub-branch that estimates the attractive force $\mathbf{f}^o$. Our strategy for choosing the goal position (and, hence, $\mathbf{e}^d$) is described later in Section IV. In order to reduce the learning complexity, we neglected the compression and the sliding friction forces, both in the SFM simulations and in the neural network, since they only play a role during contacts (which should be avoided by design of the planned path). The second sub-branch of the network then comprises a Lambda layer (followed by two single neuron layers with no bias) that takes as inputs the distance d_k and the components of the unit vector $\mathbf{n}_k$ at time t. The inputs are combined in an exponential form as in (3), where the only two learnable weights reflect the A, B parameters of the SFM. The formulation of the total force $\mathbf{f}$ in the structured NN form, as shown Fig. 1-b, is then given by

$$\begin{aligned}\mathbf{f} = \text{sig}\left(\mathcal{V}(t)W_v\right) w_{vs}\mathbf{e}^d(t) - \tanh\left(\Delta\mathbf{p}(t)W_{\mathbf{v}}\right) \odot w_{\mathbf{v}s}& \\ + \left(w_A e^{d_k(t)/w_B}\mathbf{n}_k(t)\right) \odot w_{\mathbf{f}s},& \quad (7)\end{aligned}$$

where again $w_A, w_B \in \mathbb{R}^1$, and $w_{\mathbf{f}s} \in \mathbb{R}^{1\times 2}$ are the new 4 learning weights.

IV. MULTI-GOAL PREDICTION

The branch *Net2* described in the previous Section takes as input the position of the goal. Our strategy is a multi-goal inference consisting of two steps. The first step is to formulate several hypotheses on the goal and carry out a prediction based on *Net2* for each. The second step is to perform a likelihood analysis based on the Multiple Model Approach (MMA) presented in [32]. We apply both the first-order

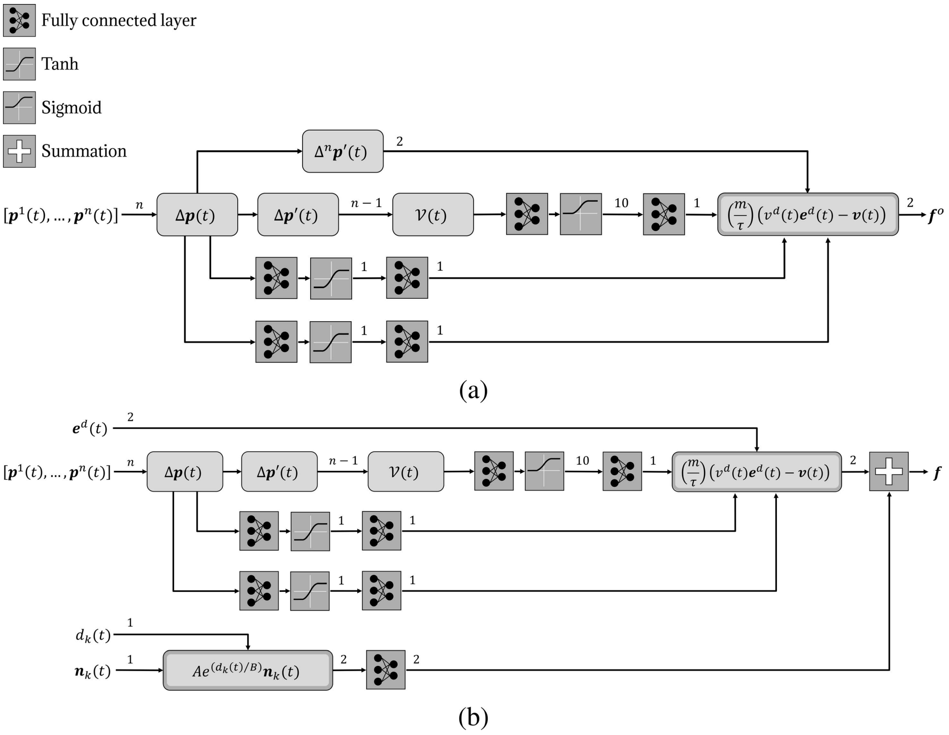

Fully connected layer
Tanh
Sigmoid
Summation
$\Delta^n \boldsymbol{p}'(t)$
2
$[\boldsymbol{p}^1(t), \ldots, \boldsymbol{p}^n(t)]$
n
$\Delta \boldsymbol{p}(t)$
$\Delta \boldsymbol{p}'(t)$
$n-1$
$\mathcal{V}(t)$
10
1
$\left(\frac{m}{\tau}\right)(v^d(t)\boldsymbol{e}^d(t) - \boldsymbol{v}(t))$
2
$\boldsymbol{f}^o$
1
1
1
1
(a)
$\boldsymbol{e}^d(t)$
2
$[\boldsymbol{p}^1(t), \ldots, \boldsymbol{p}^n(t)]$
n
$\Delta \boldsymbol{p}(t)$
$\Delta \boldsymbol{p}'(t)$
$n-1$
$\mathcal{V}(t)$
10
1
$\left(\frac{m}{\tau}\right)(v^d(t)\boldsymbol{e}^d(t) - \boldsymbol{v}(t))$
2
$\boldsymbol{f}$
1
1
1
1
$d_k(t)$
1
$\boldsymbol{n}_k(t)$
1
$Ae^{(d_k(t)/B)}\boldsymbol{n}_k(t)$
2
2
(b)

FIGURE 1. Schemes of the *Net1* (a) and the *Net2* (b). The numbers on the connections between the layers represent their output size.

generalised pseudo-Bayesian (GPB1) estimator and the Interactive Multiple Model (IMM) estimator to each predicted trajectory and select the one with the highest confidence, i.e., the most probable goal.

A. SELECTION OF THE GOALS

As the person moves across the environment, s/he chooses the next goal within an area defined as Area of Immediate Interest (AoII). Roughly speaking, the AoII is the region of the space that contains all the entities that are likely to influence a human's motion at the current time and in the near future. The AoII contains the possible goals and is estimated in a different ways depending on our prior knowledge of the environment.

Under the observation that humans move preferably headways [6], it is reasonable to expect that the motion is attracted by an entity inside a disk sector with aperture lower than 180°. Our experiments suggested that a good choice for the aperture is $\theta = 160°$, which incidentally is an approximation of the human field of view, including binocular and peripheral vision [33]. The radius of the sector is related to the distance that can be travelled in the interval time corresponding to the prediction horizon.

Considering that the possible goals cannot be within an obstacle, we can estimate the AoII by intersecting the disk sector with an occupancy grid derived from the map (see the shaded area in Fig. 2). In formal terms the grid map, composed of N cells, is denoted as $\mathcal{C} = \{c_1, \ldots, c_N\}$, whose elements are labelled with 0 if occupied by an obstacle, 1 otherwise. The AoII at time t is denoted by $\mathbf{c}(t) \subseteq \mathcal{C}$ and is defined as:

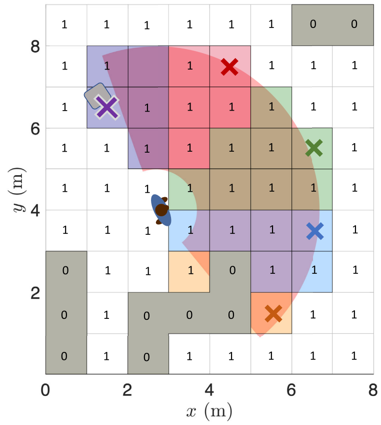


FIGURE 2. Pedestrian's AoII and the cells grouped according to (8). The cells are labelled with 0 if occupied by an obstacle, 1 otherwise. The coloured crosses represent the selected goals. The AoII is estimated first removing the obstacles from the disk sector (shaded area), then it is split into different sub-sectors and for each of them we select a candidate either on a point of interest (purple cross) or via geometric considerations.

$$\mathbf{c}(t) = \{\forall c \in \mathcal{C} : [\text{Disk}(t) \cap c] \neq \emptyset \land c = 1\}, \qquad (8)$$

where Disk(t) is the disk sector described above centred in the position of the human.

The AoII is segmented in $K-1$ equally spaced radial cones with aperture ϑ, each of them representing a different area of interest. With an odd number of cones, we have one cone associated with the human's decision to walk forward, while the others represent potential turns. Similarly to (8), we define the k-th (for $k = 1, \dots, K-1$) cone cells as $\mathbf{c}_k(t) \subset \mathbf{c}(t)$. For each $\mathbf{c}_k(t)$ we select a goal $g_k(t)$ according to the following logic. If a known point of interest falls within $\mathbf{c}_k(t)$ we set $g_k(t)$ equal to the point of interest, otherwise $g_k(t)$ is given the cell centroid maximising the Manhattan distance to the pedestrian occupied cell ($c_{\mathbf{p}}(t)$) and minimising the angular displacement from the k-th cone bisecting direction ($\psi_k(t)$), i.e.

$$g_k(t) = \arg\left(\max_{c\in\mathbf{c}_k(t)} ||c - c_{\mathbf{p}}(t)|| \min_{c\in\mathbf{c}_k(t)} \angle\overline{c\, c_{\mathbf{p}}(t)} - \psi_k(t)\right).$$

In cases where the map of the environment is not available or is incomplete, the occupancy grid could not reveal the position of some obstacles (i.e., we could have some cells labelled as 1, whereas they should be labelled as 0). As a consequence, we could select some candidate goals in unrealistic or impossible positions. This is not a critical problem because the estimation technique described below will quickly deplete the likelihood of fictitious goals. Still, the use of additional information enables us to purge unrealistic candidates and, hence, leads to a better performance. For instance, the robot's sensors could reveal the presence of some of the obstacles despite their limited visibility of the scene allowing us to correctly label as 0 at least some of the occupied cells.

To model the a-priori probability of selecting one of the goals we adopt the von-Mises distribution, that is a Gaussian pdf with mean $\phi(t)$ (the pedestrian actual heading), variance σ^2 and normalised in $[-\pi, \pi]$, denoted with $\mathcal{N}_{VM}(\phi(t), 1/\sigma^2) = \exp(\kappa\cos(\vartheta_k - \phi(t)))/(2\pi I_o(\kappa))$, where κ is the concentration parameter, and $I_o(\kappa)$ is the modified Bessel function. In practice, the confidence of the k-th goal belonging to the cone with aperture ϑ_k is given by

$$\Pr\left[g_k(t)|\mathbf{p}^t\right] = \int_{\vartheta_k} \mathcal{N}_{VM}(\nu; \phi(t), 1/\sigma^2)d\nu.$$

All the goals described above represent the human's intention to continue walking, preferably forward. To model the intentions to stop, or eventually make a ''U turn'', we add one goal $g_K(t)$ placed in the current pedestrian position $\mathbf{p}^t$. To set the confidence of $g_K(t)$, we consider an heuristic value equal to half the $\min_{k=1,\dots,K-1} \Pr\left[g_k(t)|\mathbf{p}^t\right]$.

While the agent moves in the environment, the goal positions are updated in order to be compliant with the current pedestrian location. Specifically, knowing the set of goals at $t-1$, we first compute the AoII geometry at t obtaining a new set of goals, then we solve the proper association between $g_k(t-1)$ and $g_k(t)$ via the Munkres assignment algorithm [34].

B. LIKELIHOOD COMPUTATION

For each possible goal we compute a *Net2* prediction and execute a Kalman Filter (KF) iteration. The prediction step for the k-th goal is given by

$$\begin{aligned}\bar{\mathbf{s}}_k^{t+1} &= \mathcal{F}(\bar{\mathbf{s}}_k^t, \mathbf{f}),\\ \Sigma_k^{t+1} &= A\Sigma_k^t A^T + BQB^T,\end{aligned} \tag{9}$$

where $\bar{\mathbf{s}}_k^t = \left[\bar{\mathbf{p}}_k^t, \bar{\mathbf{v}}_k^t\right]^T$ is the state at time t, $\mathcal{F}(\cdot)$ represents the second-order dynamic model (1), and $A = \frac{\partial\mathcal{F}(\bar{\mathbf{s}}_k^t,\mathbf{f})}{\partial\bar{\mathbf{s}}_k^t}$ is the linearised system dynamic matrix. Moreover, Σ_k^t is the covariance matrix of the estimation error associated with the k–th goal state, $B = \frac{\partial\mathcal{F}(\bar{\mathbf{s}}_k^t,\mathbf{f})}{\partial\mathbf{f}}$ is the force linearised input vector, having additive uncertainties with covariance matrix Q (that we assume to be proportional to a perturbation in the acceleration space). In the update step, we use the observations $\mathbf{p}^{t+1}$ to evaluate the innovation vector $\boldsymbol{\varepsilon}_k^{t+1}$, its covariance S_k^{t+1}, and update the state covariance, i.e.

$$\begin{aligned}\boldsymbol{\varepsilon}_k^{t+1} &= \mathbf{p}^{t+1} - H\bar{\mathbf{s}}_k^{t+1},\\ S_k^{t+1} &= H\Sigma_k^{t+1}H^T + R,\\ K_k^{t+1} &= \Sigma_{i,k+1}^{-}H^T(S_k^{t+1})^{-1},\\ \Sigma_k^{t+1} &= \Sigma_k^{t+1} - K_k^{t+1}H\Sigma_k^{t+1},\end{aligned} \tag{10}$$

where R is the covariance matrix of the measurement uncertainty and $H = [I, 0]^T$ is the output matrix. The computation of the likelihood can be carried out through two different approaches: GBP1 and IMM [32]. The former is easy to set up but has a good performance only when the choice of the goal is relatively consistent in time (i.e, the human does not change her/his mind on where to go). The latter caters for possible changes in the goal through an homogeneous Markov Chain (MC). The higher level of generality of IMM requires some additional effort in the calibration of the transition probabilities in the MC.

When using GPB1, the probability of $g_k(t+1)$, dubbed μ_k^{t+1}, can be estimated using the following Bayesian rule:

$$\mu_k^{t+1} = \Lambda_k^{t+1}\mu_k^t/\Pr\left[\mathbf{p}^{t+1}\right], \tag{11}$$

where $\Lambda_k^{t+1} = \Pr\left[\mathbf{p}^{t+1}|g_k(t+1)\right]$ is the likelihood of the observed position $\mathbf{p}^{t+1}$ for the goal $g_k(t+1)$, which can be modelled as a bivariate Gaussian pdf with mean $\boldsymbol{\varepsilon}_k^{t+1}$ and covariance S_k^{t+1}. Unlike the standard GPB1 approach, where the state estimates for each model (i.e. goal) are fused together, we let each KF run independently, therefore the updated states are not needed.

In the more general case in which we use the IMM approach, we combine the goal probabilities μ_k^{t+1} in (11) with the transition probabilities p_{kj} of dynamically changing the goal, i.e. $p_{kj} = \Pr\left[g_k(t+1)|g_j(t)\right]$. The goal switching is assumed to be a homogeneous Markov process, so that the transition probabilities are known and time-invariant. Hence, in the IMM, (11) is substituted with the mixing probabilities

$$\mu_k^{t+1} = \frac{\Lambda_k^{t+1}}{\sum_{k=1}^{K}\Lambda_k^{t+1}\sum_{j=1}^{K}p_{kj}\mu_j^t}\sum_{j=1}^{K}p_{kj}\mu_j^t. \tag{12}$$

The Markov chain transition probabilities are defined as a stochastic matrix. The persistence of each mode is equal to a probability α, while the transition towards each other mode is an even distribution of $1-\alpha$, i.e.

$$p_{kj} = \alpha I + [(1-\alpha)/(K-1)]\,(1-I), \tag{13}$$

where 1 is a matrix with all ones and proper dimensions.

As the agent moves in the environment, we eventually repeat the *Net2* predictions based on the new measured human position, and the updated goals. This occurrence is triggered by one of the following events: *i*) the average among all the innovations (i.e., the norm of $\boldsymbol{\varepsilon}_k^t$ in (10)) is greater than an upper limit ε_M; *ii*) the Euclidean distance between the observation $\mathbf{p}^t$ and one goal $g_k(t)$ is lower than δ_M (i.e. the pedestrian is close to the k-th goal). When any of these conditions befalls and we reinitialise $g_k(t)$ to $g_k^\star(t)$, its first confidence $\mu_k^{t,\star}$ is obtained using μ_k^t as a prior, i.e.

$$\mu_k^{t,\star} \leftarrow \lambda\mu_k^t + (1-\lambda)\Pr\left[g_k^\star(t)|\mathbf{p}^t\right], \tag{14}$$

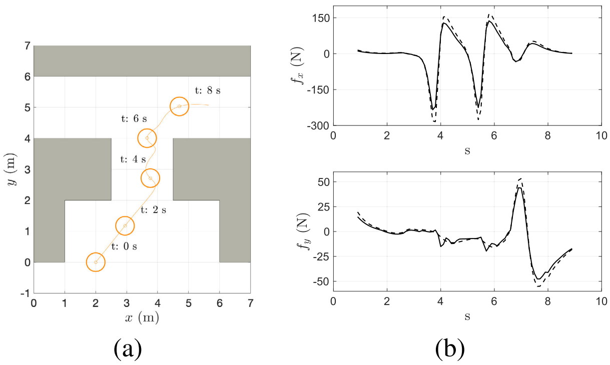
where $\lambda \in [0,1]$ is a weighting parameter. The rationale is that the probabilities reached before the reinitialisation can hold as a starting guess, thus we mix them with the a-priori $\Pr\left[g_k^\star(t)|\mathbf{p}^t\right]$.

V. TRAINING AND EXPERIMENTAL VALIDATION

In this section we show a number of experimental results reported on well known datasets and on other data sets of our making. We first describe the training phase, which was totally conducted using synthetic data, and then we will discuss the performance of our predictor in different operating conditions.

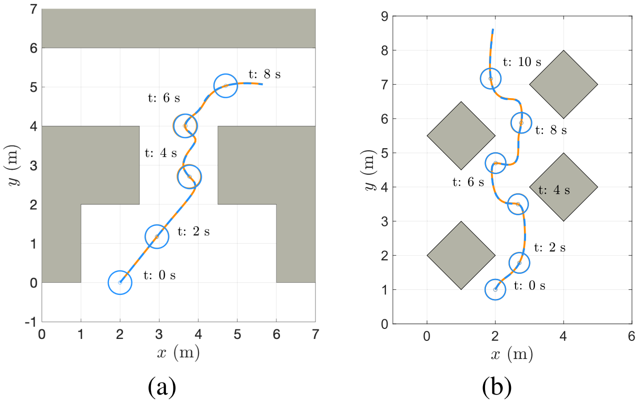
A. TRAINING DETAILS

Both *Net1* and *Net2* were implemented in Keras and trained with the Adam optimiser with a learning rate of 0.005, batch size 128, and number of epochs 300 using a 2.7 GHz Intel Core i7 processor. We created two synthetic sets of trajectories generated by the SFM in an open space (for *Net1*) and in a structured environments (for *Net2*). For the structured case, the environment we assumed consisted of two intersecting corridors (as in Fig. 5-a). In the first set, we ran 800 simulations of 20 seconds with a sampling time of $\delta_t = 0.1$ s. The initial positions were randomly chosen within a range between 8 and 10 m from the final goal, and, for each simulation, a random set of parameters for the SFM model in (2) were taken from the intervals [50, 90] kg for m, [0.5, 0.9] s for τ and [0.5, 3] m/s^2 for v^d. For the second set, we ran 1200 simulations where the agent moves through the corridors intersection, starting from one of the four possible waypoint areas (see Fig. 5-a) and reaching another one. We set the parameters in (3) to $A = 1000$, $B = 0.08$, according to [5]. The window of the motion observations was empirically set to $n = 10$ steps, which provides a good trade off between learning speed and network prediction accuracy without over-fitting. This is consistent with the fact that the networks mostly depend on the most recent data, and that a longer motion observation does not significantly improve the prediction accuracy [20].



(a) (b)

FIGURE 3. Validation example. (a) SFM simulated trajectories and (b) *Net2* forces predictions (solid) compared with the SFM forces (dashed).



(a) (b)

FIGURE 4. Comparison of actual (solid) and predicted (dashed) trajectories using learned *Net2* forces with $n = 10$ steps with a prediction horizon of (a) 7 and (b) 9 seconds in a structured environment.

The 70% of each synthetic dataset was used as the training sets, while the remaining samples were used for validation. Notice that, in order to avoid possible correlations between training and validation, the samples randomisation was done after dividing the two sets. In Table 1 we report the hyperparameters and the implementation characteristics of our proposed networks *Net 1* and *Net 2* and other comparative models used in the experiments in Sections V-C and V-D.

B. VALIDATION ON SIMULATED DATA

After the training, we performed a first validation step on a different batch of simulation data. The difference between this batch and the one used for training lies in the geometric configuration of the simulated environment (which was no longer a simple intersection of two corridors).

1) SFM FORCES GENERATION

For this evaluation, we first generated the entire trajectory. For each step, the network was used to infer the force sample. In Fig. 3 we report the results of the force predictions of the *Net2* network (similar results are obtained for *Net1*). Fig. 3-a depicts the trajectories generated by the SFM, while Fig. 3-b the comparison between the real and the network predicted force components $f_x = f_x^o + f_x^w$ and $f_y = f_y^o + f_y^w$ described in (2) and (3). Despite a slight underestimation of the forces in the occurrence of the horizontal collisions with the walls (see the peaks in Fig. 3-b), the prediction remains consistent even if the configuration of the obstacles was very different from the one used in training (see Fig. 3-a). We can legitimately conclude that the performance of *Net2* has not been negatively affected by the environmental bias introduced during the training phase. The structure of our NN and the abstract modelling of the environment guides the learning process toward a correct understanding of the "physics" of the human motion rather than of specific behaviours, and makes the learning results usable across a wide gamut of possible environments. This fact is further substantiated by the inspection of the learned weights. In our example, the weights of the second branch $w_A w_{\mathbf{f}s} = 3.4 \cdot 10^3$ and $w_B = 0.0834$ are pretty close to the A and B parameters, respectively, used in the synthetic dataset. Likewise, if we combine the weights of the first branch, as in (7), we obtain an estimated mass of 55.6 kg, which is correctly positive and of the same order of magnitude of the chosen m. Since the NN internal weights correspond to physical quantities, unexpected values (e.g., negative masses, exaggerated velocities) are used to reveal either the absence of a real convergence or that the system has been over-fitted. This "interpretability" of the results is the most important trait of the future generation of explainable AI systems [35].

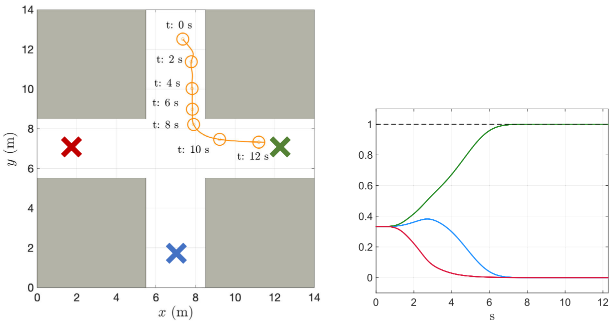
2) LONG TERM PREDICTION

Given that the forces are correctly estimated, we fed those quantities into the SFM (1) to generate long term motion predictions. To this end, we first collected one second of measurements from a trajectory corresponding to a window of $n = 10$ samples. These measurements were then used to estimate of the SFM force, and, hence, the position at the next step. The whole procedure was iteratively repeated shifting each time ahead by one sample the window. This way, after 10 prediction steps, the computation was made "in open loop", relying solely on the SFM predictions. The comparison between the predicted trajectory and the actual one is shown in Fig. 4 for two different scenarios. In both scenario, the actual trajectories (solid lines) are well replicated by the predicted human motions (dashed lines) even for a long horizon (7 or 9 seconds, respectively).

3) THE CASE OF UNCERTAIN GOALS

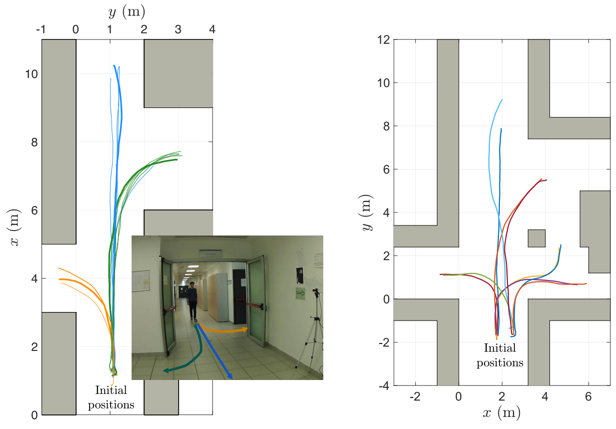
In the example described above, our objective was to validate the NN based prediction of the human motion. For this reason, we assumed that the goal of the human target was known. Now, we move to the general case of unknown goal in order to validate the entire approach.

Let us consider again the intersecting corridors scenario. Following the method described in Section IV, we identify four goals, which correspond indeed to the areas chosen for the network training. The pedestrian starts from one of the areas (from the uppermost one in the experiment in Fig. 5-a), and reaches one of the other three. Therefore, we generate three different hypotheses for the trajectory predictions with the *Net2* network (one per area, respectively) choosing as goal the area centroids. As shown in Fig 5-b, after about 2 s, the GBP1 classifier is able to find the correct goal, while in the following seconds the confidence towards the simulated trajectory increases. The evaluation of *Net2* with the multi-goal strategy is then further proved on real human trajectories collected in a structured environment. In particular, we record the data in two different portions of a hallway with multiple exits in our department at the University of Trento. Data were collected using a LiDAR, which was placed about 90 cm from the ground at the centre of the two scenes, in order to entirely see every side of the corridors (see Fig. 6-a). The sensed data were used to both extract the walls information (that is, the static points between subsequent frames) and the pedestrian observations. Our acquisition algorithm was used to extract points belonging to the person's waist, and clustered



(a) (b)

FIGURE 5. (a) The agent moves from the uppermost area to the green exit on the right. The coloured crosses represent all the possible goals of the scenario. (b) Probabilities of the three goals estimated by the GPB1 method.

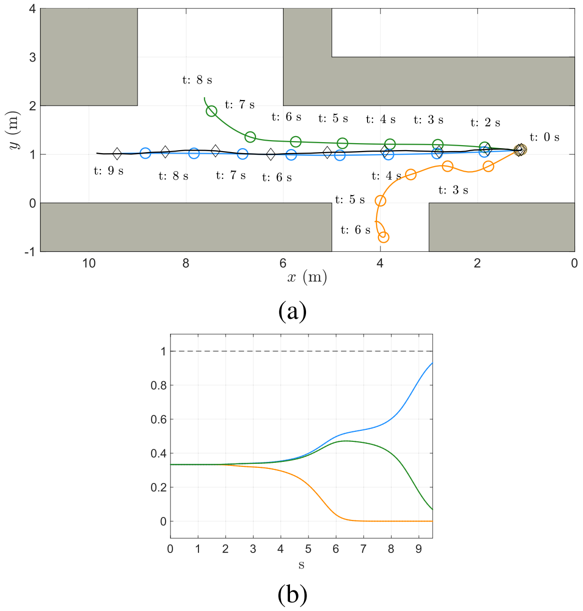


(a) (b)

FIGURE 6. Experimental trajectories in a hallway. (a) First set of collected trajectories: bold lines are the ones used for the multi-goal classifier evaluation and a picture of the experimental set-up. (b) Second set of collected trajectories.

TABLE 1. Comparison of implementation details of our proposed neural networks and related models used in the experiments reported in Sections V-C and V-D. Specifically, we compare our models *Net1* and *Net2* (SFM-NNs), three different Fully Connected neural networks (FCs), the Feed Forward neural network (FF) and the LSTM network (LSTM) from [20]. The speed of execution was measured by running 10000 times over a prediction window of 4.8 s: (a) the Keras implementation of each neural network (reported in seconds), (b) the C++ implementation of the same networks (reported in microseconds).

		SFM-NN (*Net1*)	SFM-NN (*Net2*)	FC 1/2/3	FF	LSTM
Prediction	Number of parameters	123	127	22,080 / 12,380 / 9,560	2694	201,240
	Inference speed (Keras version)	0.1484 s	0.1823 s	0.0012 s / 0.0015 s / 0.0680 s	-	-
	Inference speed (C++ version)	60.7050 μs	70.6534 μs	65.4340 μs / 41.0731 μs / 1478.07 μ	-	-
Training	Learning rate	0.005	0.005	0.005	0.0004	0.0004
	Epochs	300	300	300	35	35
	Batch size	128	128	128	64	64
	Optimizer	Adam	Adam	Adam	Adam	Adam



(a)

(b)

FIGURE 7. (a) Trajectory predictions compared with the ground truth (black line). (b) Probabilities of the trajectory predictions while the pedestrian moves towards the blue goal, as estimated by the GPB1 method.

them into a single planar position. In the first recorded set, depicted in Fig. 6-a, the person could go to three different goals, i.e. one directly to the left, one to the right and one right at the end of the hallway (see the captured inlet image). In Fig. 7-a we report the classification result for the trajectory of the *blue* goal with the GPB1 estimator. In this experiment, we manually give to the model the prior knowledge of all the goal locations, i.e. the beginning of the corridors. As reported in Fig. 7-b, the first goal on the left of the person is discarded after about 4 s, while the confidence of the two remaining goals remains almost the same, until the correct goal is found after about 6 s, before the pedestrian oversteps the next exit. Notice that the oscillating trajectory of the orange sample is due to the SFM dynamic with respect to static obstacles.

C. *Net2* PERFORMANCE EVALUATION

To experimentally validate the performance of the multi-goal strategy with the IMM estimator, we used the second set of actually recorded trajectories in Fig. 6-b, where the person could go towards different exits, located at different walking directions. Wall shapes were restored by looking at the static measures of the LiDAR, while here we use no prior information on the points of interest positions. As customary in the literature from this kind of problems [9], [20], [36], we compute the errors using the *Mean Euclidean Distance* (MDE) and the *Final Displacement Error* (FDE$_\mathrm{K}$) to determine the approach performance, i.e.

$$\begin{aligned} \mathrm{MDE}_K &= \min_{i\in\{1,\dots,K\}} \frac{1}{n_f} \Sigma_{j=t+1}^{t+n_f+1} \left\| \mathbf{p}^j - \mathbf{p}_i^j \right\|_2, \\ \mathrm{FDE}_K &= \min_{i\in\{1,\dots,K\}} \left\| \mathbf{p}^{t+n_f+1} - \mathbf{p}_i^{t+n_f+1} \right\|_2, \end{aligned} \tag{15}$$

where n_f is the number of sample of the predicted trajectory. In particular, due to the presence of multiple forecasted future trajectories, we select the best predicted trajectory among the K samples. In the literature, the observation window is usually set to 8 time-steps (that is, 3.2 s according to the data acquisition frame rate of the datasets), while the predictions span the successive 4.8 s. In our model (**SFM-NN**), we observe only for 1 s (n=10 steps) of the real-world trajectories and predict for 4.8 s for a fair comparison. Our strategy was compared with other two orthogonal approaches. We implemented the Constant Velocity model (**CV**) as proposed by [20], where the multimodality is introduced by predicting $K = 20$ samples with a random noise on the walking direction, i.e. $\mathcal{N}(0, \sigma_a^2)$, with $\sigma_a = 25°$. Moreover, we trained different Fully Connected neural networks with the same hyper parameters of *Net2* with the synthetic dataset. For each network we tested several structures in order to find the best fitting. The **FC1** is made of two hidden layers with 100 and 80 neurons, both followed by ReLU activation functions. It takes as input all the past motions and has two output layers with 48 neurons each to return all the prediction steps. The **FC2** receives as inputs also explicit scene information (the same way as *Net2* does), has two hidden layers with 50 and 30 neurons, and predicts for the entire window. The **FC3** is similar to the **FC2**, however as well as our proposed approach, it predicts only the position at $t+1$ and recursively uses the past predictions as input for the new inference. In Table 2 we report the prediction errors for all the evaluated models. The proposed **SFM-NN** masters the additional complexity of the scene by using *Net2*, while the influence of the environment is implicitly learned by the **FC1** and explicitly encoded in the **FC2** and **FC3**. Nonetheless,

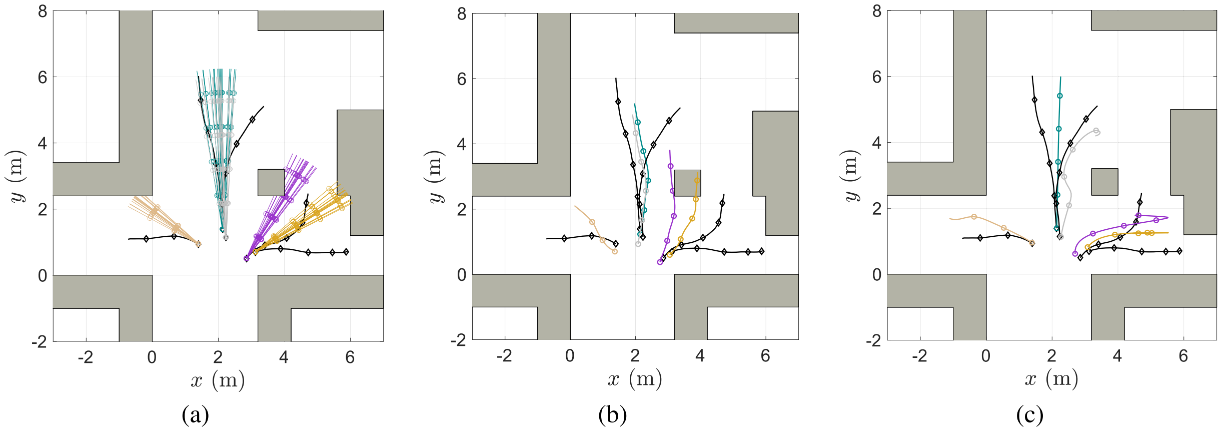


FIGURE 8. Predicted trajectories with CV (a), FC1 (b) and SFM-NN-s (c). The ground truth is depicted in black for the five sampled trajectories, while colour codes are adopted for the corresponding predicted trajectories from the different approaches.

TABLE 2. Comparison of prediction errors with our model (SFM-NN), Constant Velocity (CV) model and Fully Connected (FCs) networks, on collected trajectories. The suffix "-s" depicts the models with only the most confident estimate. The metrics are reported in meters.

	K	MDE$_K$	FDE$_K$		MDE	FDE
CV	20	0.47	1.28	CV-s	1.65	2.73
FC1	1	0.96	1.84	SFM-NN-s	**0.74**	**1.49**
FC2	1	0.96	1.97			
FC3	1	0.65	1.43	SFM-NN cl	**0.16**	**0.16**
SFM-NN	10	**0.38**	**0.68**			

our approach outperforms all the other approaches w.r.t. both MDE and FDE and even if we drew just $K = 10$ samples. Therefore, even if the information of the environment are considered, the classical NN approaches are strongly scenario-dependent and poorly generalise when trained with synthetic datasets. Unlike other multimodal approaches, the advantage of our model is to have the a-priori confidence estimated over the different modes as depicted in Section IV. As shown in Table 1, the **FC**s obtained the fastest execution times on the Keras implementation, in particular the **FC1** and **FC2** since both are designed to infer the whole prediction window with a single execution. The peculiar structure of our networks makes them slightly slower in their Keras implementation, however, this is not a significant disadvantage as it is used for offline prediction. In the C++ implementation (introduced in Section VI-A), the inference times are suitable for real-time robotic applications.

To increase the fairness, we also endowed the **CV** model with an IMM estimator to obtain a-priori confidence as well. For comparison, we select for both the algorithms the most probable trajectory, i.e. the one with the highest confidence, and named them **CV-s** and **SFM-NN-s**, respectively. This way, in the prediction phase both models were reduced to single-modal approaches. The comparative results reported in Table 2 further confirm the effectiveness of our solution. For a visual representation of the behaviour of the different approaches in Table 2, we offer in Fig. 8 a qualitative representation of the different predicted trajectories. The **CV** model (Fig. 8-a) reached a good average performance. However, walls information and trajectory curvature changing are not obviously predictable based on the motion history. The **FC1** (Fig. 8-b) properly predicts linear trajectories, while it was not able to catch the turning behaviour nor the environmental effects. Instead, even if only the most probable trajectory is predicted (i.e., **SFM-NN-s**), our model (Fig. 8-c) correctly predicted all the walking directions and well approximated the pedestrian behaviours.

Finally, in the last column of Table 2 we reported the performance of the IMM estimator in closed loop, that is with the continuous update of the goal probabilities with respect to the observations, i.e. we let the *Net2* behave as a filter when new observations come. These experimental results are of paramount relevance for an application of the solution to real-time pedestrian motion estimation of service robots. Notice that in this case, the performance are definitely increased and the two metrics, MDE and FDE, are obviously the same, i.e., we come up with just one prediction.

D. *Net1* PERFORMANCE EVALUATION

To further substantiate the analysis, we experimentally validate the performance of the *Net1* network and make a comparison with other methods in the literature, we used two widely known human motion datasets: the ETH [37] dataset (with the scene *Hotel* and *ETH*) and UCY [38] dataset (with scene *UCY*, *Zara1* and *Zara2*). These datasets comprise real world human trajectories in open scenarios, where the influence of static obstacles is mostly negligible, thus the embedded SFM model plays a major role. The performance were evaluated using the MDE$_K$ and the FDE$_K$ defined in (15), where $K = 1$ since we are considering for *Net1* just one model. Unlike the leave-one-out approach used in [9], [36], we used the synthetically learned *Net1* to validate the prediction accuracy over real-world data, and we use the same observation and prediction window as in Section V-C, In Table 3 we show the prediction errors comparison with the Constant Velocity (**CV**), the Constant Accelerated (**CA**) models and the Feed Forward (**FF**) neural network, all implemented by [20]. Moreover, we report the comparison with the LSTM network

TABLE 3. Prediction errors (average errors in AGV rows) with our model (SMF-NN) and other state-of-the-art models on the real-world datasets. The metrics are reported in meters.

Dataset	Metric	CV	CA	FF	LSTM	SFM	SFM-NN
Hotel	MDE	0.27	0.95	1.59	**0.15**	0.69	0.34
	FDE	0.51	2.41	3.12	**0.33**	1.63	0.75
ETH	MDE	**0.58**	1.35	0.67	0.60	0.89	0.61
	FDE	**1.15**	3.29	1.32	1.31	2.12	1.48
UCY	MDE	0.46	0.79	0.69	0.52	0.89	**0.42**
	FDE	1.02	2.03	1.38	1.25	2.12	**1.02**
Zara1	MDE	0.34	0.59	0.39	0.43	0.61	**0.31**
	FDE	**0.76**	1.50	0.81	0.93	1.43	0.78
Zara2	MDE	0.31	0.50	0.38	0.51	0.84	**0.27**
	FDE	0.69	1.30	0.77	1.09	1.96	**0.67**
AVG	MDE	0.39	0.84	0.74	0.44	0.79	**0.39**
	FDE	**0.83**	2.11	1.48	0.98	1.85	0.94

(**LSTM**) by [36], and with the standard isolated SFM model (**SFM**) with its parameters optimised with the leave-one-out approach on the real dataset.

At a first look, our model still shows good prediction performance in each dataset, with an average MDE of about 40 cm with the exception of the *ETH* scenario. The performance decrease in the case of the final displacements measured by FDE: this result is mainly due to the strong non-linearity of the real-world trajectories, which are not easily followed by *Net1* in open loop. However, the results are instead remarkable for the following reasons: 1. compared with the optimally trained **SFM**, the results are radically better due to the presence of the NN that is able to add flexibility to the predictions coming from the context; 2. notwithstanding the worsened training conditions of the proposed approach, the reduced observation interval and the absence of obstacles (which are the main novelty of the proposed approach, as aforementioned), our results provides similar performance than the other methods in the literature; 3. the proposed method, despite having similar performance as state-of-the-art approaches (e.g., the **CV**) in open spaces, has the potential to further model other effects, such as human to human interactions, which will be the subject of future investigations.

VI. ROBOTIC IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this section we show a set of experiments conducted using a real robot. We first describe the platform's components and the software framework, and then we will discuss the performance of the proposed solution.

A. ROBOTIC PLATFORM

In order to create a suitable robotic platform for the experiments, we equipped a two-wheeled unicycle robot with a sensing system composed of a 2D Light Detection And Ranging (LiDAR) and an RGB-D camera (see Fig. 9-a). The RGB-D sensor is used for human detection and tracking, while the LiDAR data are used to improve the tracking performance, to localise the robot in the environment and to measure the distances to the surrounding obstacles. The LiDAR (an RPLidar A3[1]) has a view angle of 360°, a maximum measuring distance up to 40 meters, and is typically operated at 20 revolutions per second. The RGB-D camera adopted is an Intel® RealSense™ D435,[2] working in an ideal range spanning from 0.5 to 3 m. In addition, rotational encoders on each rear wheel and a visual inertial camera facing upwards (an Intel® RealSense™ T265[3]) provides odometry data.

The solution proposed in this paper was prototyped in C++ and is comprised of two modules (see Fig. 9-b): *detection and tracking* and *motion prediction*. The algorithm was executed on a Jetson TX2 and on a Intel® NUC, both embedded on the robot.

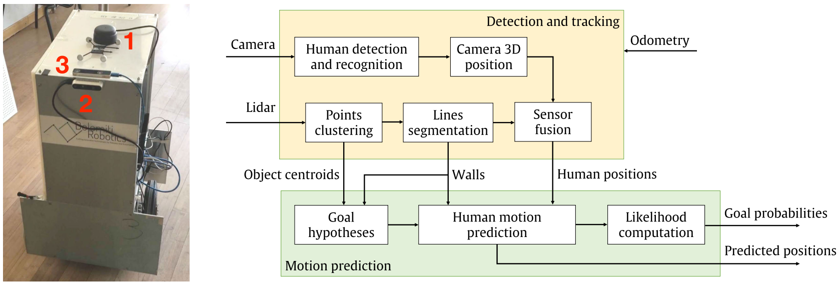
The *detection and tracking* module is used to: 1. provide real-time information on the presence of fixed obstacles and walls, 2. detect and track the human target. The human tracking algorithm is derived from our previous work [39], in which we first identify and track the person in the image space of the camera, and then we merge the camera data and the LiDAR readings in order to identify the 2D position of the human target even in presence of occlusions and/or of other moving entities. The origin of the fixed reference frame $\langle F\rangle$ can be freely defined in the available map or w.r.t. the robot's initial position. The subsequent positions of the robot $\mathbf{p}_r = [x_r, y_r, \phi_r]^T$ are then retrieved with a standard localisation algorithm [40]. The assumed sensing configuration allows us to bring all the measurements taken by the robot in its relative reference frame $\langle R\rangle$ back to the fixed frame $\langle F\rangle$ by subtracting the relative robot motion from the measures. The pedestrian's 2D positions are retrieved by the robot with the sensor fusion approach of [39]. As regards the detection of obstacles walls and fixed obstacles, we use the sequence of measurements from the LiDAR and apply the following steps to reconstruct the information of interest: 1. we group the points obtained from the LiDAR into clusters, 2. we filter out spurious points through Ramer-Douglas–Peucker algorithm [41], 3. we interpolate the remaining points and generate a 2D evaluation of walls and obstacles boundaries. We observe that this information is key to the computation of the repulsive forces in the SFM-based prediction.

The *motion prediction* module implements the whole set of algorithmic solutions proposed in this paper to predict the motion of the human target: generation of the goal hypotheses, NN based implementation of the SFM and likelihood computation. After the network predictions, the future human positions are obtained with the double integration of the SFM model (1), with a discretization time δ_t, that is imposed by the sensor with the lowest sampling frequency and is comparable with the sampling time used in simulation. The training of the NN was performed offline and the weights produced by the training were transferred into the C++ implementation.

[1] https://www.slamtec.com/en/Lidar/A3

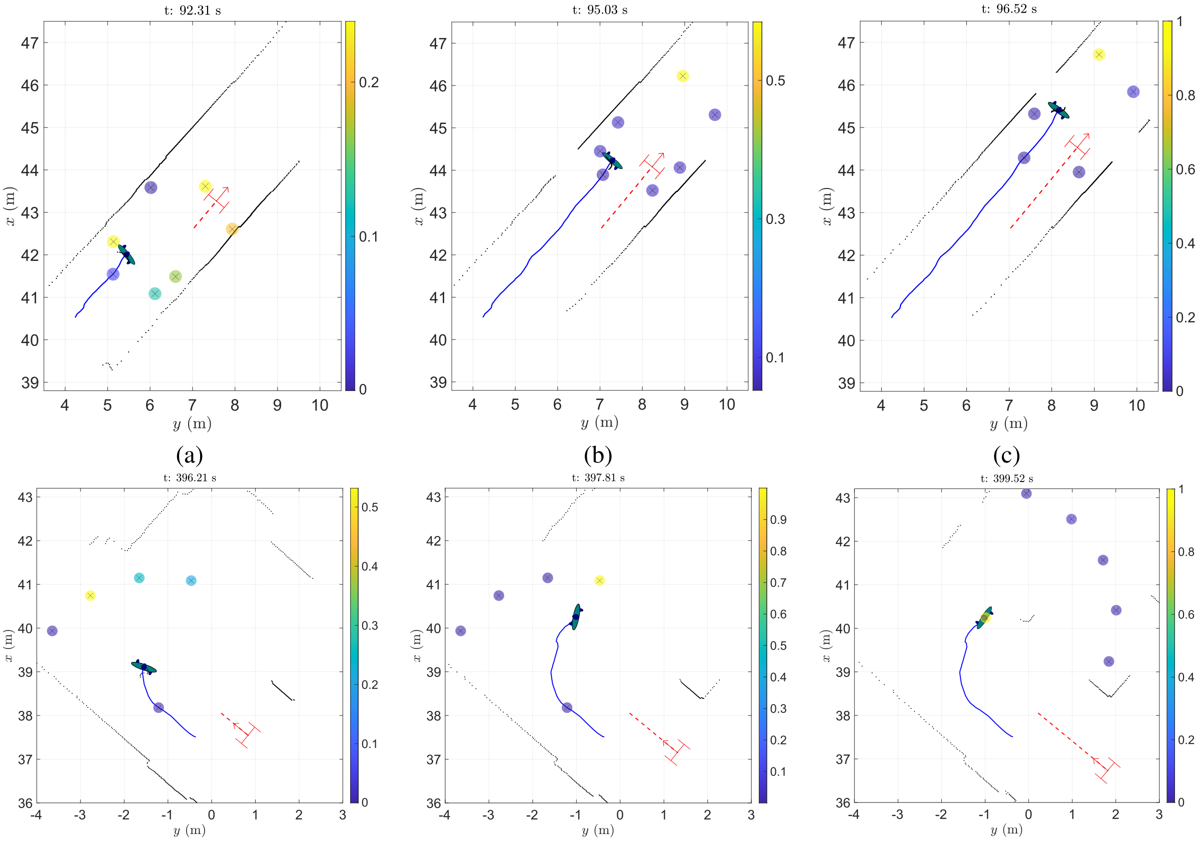
[2] https://www.intelrealsense.com/depth-camera-d435/

[3] https://www.intelrealsense.com/tracking-camera-t265/



(a) (b)

FIGURE 9. (a) Robot sensing system setup, consisting of LIDAR sensor (1) and RealSense D435 (2) for the detection and tracking, and RealSense T265 (3) for the visual odometry. (b) Scheme of the robot framework.



(a) (b) (c)

(d) (e) (f)

FIGURE 10. Different snapshots of the experimental results when a moving person (with blue solid trajectory) is tracked by a robot (red dashed trajectory). Both, a human showing a forward motion (a-c) and a turning and stopping motion (d-f) are reported. The black dots are the LiDAR data collected by the robot, while the coloured circles represent the predicted human goals at the snapshot time, each with a confidence μ_k^t depicted with the colour mapping on the side.

The network inputs (human and wall positions) are the same described in Section III-B, which are a commonplace in mobile robotics applications. Hence, the embedding of the SFM formulation in the neural network design allows us to seamlessly perform predictions both in real and simulated environments.

B. EXPERIMENTS WITH MOBILE ROBOT

The experimental evaluation of *Net2* with the multi-goal strategy is carried out through actual experiments in our department at the University of Trento. In particular, we let our robot moves and encounters people in a portion of a hallway with multiple exits. In this setting, the robot navigates in the corridors while it detects, tracks, and predicts the motion of a nearby human. The robot has no prior knowledge on the environment, nor the points of interest of the human.

Below we present qualitative results of the prediction module. All the robot collected measurements, i.e. the robot path, the human trajectory and the obstacle data, are reported in the global reference frame. As shown in Fig. 10, while the robot detects a person which is moving along the corridor in the same robot direction, the initial goal guesses are quite equiprobable (recall that the probability is measured by μ_k^t in (12)), but already catch the trend of a forward motion (Fig. 10-a with reference to the right colour code describing probabilities), while, after some additional samples of the human positions are collected, the confidence of the goal ahead is increased (Fig. 10-b) and propagated on new goals after the update described in Section IV in (14) (Fig. 10-c). In the bottom row of Fig. 10, instead, we show how a bended human path is predicted, so that the initial guess on the goal ahead (Fig. 10-d) is correctly transferred to the goal on the side (Fig. 10-e) in light of the observed trajectory and the corresponding prediction of a lateral motion. Finally, the best confidence switches to the stop-goal as soon as the person decides to remain on the spot (Fig. 10-f). These experimental results are, to the best of the Authors' knowledge, the first actual evidence of a reliable human motion prediction carried out in real-time by a moving service robot in a natural environment that deals with random human beings in an unknown scenario. We want to remark here that this is actually possible if the knowledge gained by the NN is abstracted by the wired model adopted.

VII. CONCLUSION

In this paper, we have shown a novel technique for predicting human motion embedding the environmental contextual information. Our idea is based on the combination of a neural network with a famous physics inspired dynamic model, the SFM. In the combination, each of the two approaches emphasises its own strengths and compensates for the weakness of the other. Specifically, the SFM brings a structure to the NN, reducing its complexity and the number of samples needed for the training. Furthermore, the NN predictions become explainable and physically interpretable. On the other hand, the NN expresses its full power in terms of flexibility, and of its ability to learn the complex parameter set of the SFM, which would be very difficult to estimate in real-time by conventional means for the strong non linearities of the model. Our simulations and experiments reveal the full potential of the marriage between the two worlds of physics inspired models and neural networks.

Many important points remain open and will attract our efforts in the near future. First, we plan to develop NN embedding different models which are potentially more realistic than the SFM, first and foremost the HSFM [5] and the PHSFM [42]. Second, we plan to extend the input of the model to include interaction with other humans and to account for unspoken signs (e.g., pose, eye gaze, facial expression). Third, we plan to develop motion planning algorithms designed to make the best use of learning in predicting the human motion.

REFERENCES

[1] J. P. Fentanes, B. Lacerda, T. Krajnik, N. Hawes, and M. Hanheide, "Now or later? Predicting and maximising success of navigation actions from long-term experience," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1112–1117.

[2] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg, "Real-time perception meets reactive motion generation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1864–1871, Jan. 2018.

[3] M. U. Farooq, M. N. B. M. Saad, A. S. Malik, Y. S. Ali, and S. D. Khan, "Motion estimation of high density crowd using fluid dynamics," *Imag. Sci. J.*, vol. 68, no. 3, pp. 141–155, Apr. 2020.

[4] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, p. 4282, May 1995.

[5] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo, "Walking ahead: The headed social force model," *PLoS ONE*, vol. 12, no. 1, Jan. 2017, Art. no. e0169734.

[6] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, "On the nonholonomic nature of human locomotion," *Auto. Robots*, vol. 25, nos. 1–2, pp. 25–35, Aug. 2008.

[7] A. Colombo, D. Fontanelli, A. Legay, L. Palopoli, and S. Sedwards, "Motion planning in crowds using statistical model checking to enhance the social force model," in *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 3602–3608.

[8] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli, "Reactive planning for assistive robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1276–1283, Apr. 2018.

[9] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," 2019, *arXiv:1905.06113*.

[10] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 464–469.

[11] T. Ikeda, Y. Chigodo, D. Rea, F. Zanlungo, M. Shiomi, and T. Kanda, "Modeling and prediction of pedestrian behavior based on the sub-goal concept," *Robotics*, vol. 10, pp. 137–144, Jul. 2013.

[12] T. Kretz, J. Lohmiller, and P. Sukennik, "Some indications on how to calibrate the social force model of pedestrian dynamics," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 20, pp. 228–238, Dec. 2018.

[13] I. Hasan, F. Setti, T. Tsesmelis, V. Belagiannis, S. Amin, A. D. Bue, M. Cristani, and F. Galasso, "Forecasting people trajectories and head poses by jointly reasoning on tracklets and vislets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1267–1278, Apr. 2021.

[14] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani, "Forecasting interactive dynamics of pedestrians with fictitious play," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 774–782.

[15] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[16] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1349–1358.

[17] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 759–776.

[18] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," 2020, *arXiv:2001.00735*.

[19] F. Pasquale, "Toward a fourth law of robotics: Preserving attribution, responsibility, and explainability in an algorithmic society," *Ohio St. LJ*, vol. 78, p. 1243, Jul. 2017.

[20] C. Scholler, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, Apr. 2020.

[21] Z. Wan, X. Hu, H. He, and Y. Guo, "A learning based approach for social force model parameter estimation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 4058–4064.

[22] Z. Zhang and L. Jia, "Direction-decision learning based pedestrian flow behavior investigation," *IEEE Access*, vol. 8, pp. 15027–15038, 2020.

[23] A. Johansson, D. Helbing, and P. K. Shukla, "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data," *Adv. Complex Syst.*, vol. 10, pp. 271–288, Dec. 2007.

[24] M. D. Lio, D. Bortoluzzi, and G. P. R. Papini, "Modelling longitudinal vehicle dynamics with neural networks," *Vehicle Syst. Dyn.*, vol. 58, no. 11, pp. 1675–1693, 2019.

[25] T. Kanda, D. F. Glas, M. Shiomi, and N. Hagita, "Abstracting people's trajectories for social robots to proactively approach customers," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1382–1396, Dec. 2009.

[26] J. Wu, J. Ruenz, and M. Althoff, "Probabilistic map-based pedestrian motion prediction taking traffic participants into consideration," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1285–1292.

[27] D. Karunarathne, Y. Morales, T. Kanda, and H. Ishiguro, "Model of side-by-side walking without the robot knowing the goal," *Int. J. Social Robot.*, vol. 10, no. 4, pp. 401–420, Sep. 2018.

[28] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–5.

[29] R. Senanayake and F. Ramos, "Directional grid maps: Modeling multi-modal angular uncertainty in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3241–3248.

[30] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *Int. J. Social Robot.*, vol. 7, no. 2, pp. 137–153, Sep. 2014.

[31] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 676–682, Jan. 2020.

[32] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.

[33] D. Harrington, *The Visual Fields: A Textbook and Atlas of Clinical Perimetry*, 5th ed. St. Louis, MO, USA: Mosby, Jan. 1981.

[34] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

[35] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[36] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[37] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.

[38] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[39] A. Antonucci, P. Bevilacqua, S. Leonardi, L. Palopoli, and D. Fontanelli, "Humans as pathfinders for safe navigation," 2021, *arXiv:2107.03079*.

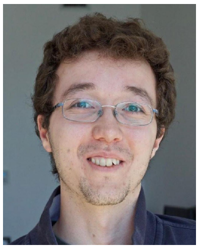
[40] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor localization of mobile robots through QR code detection and dead reckoning data fusion," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 6, pp. 2588–2599, Dec. 2017.

[41] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geograph. Inf. Geovis.*, vol. 10, no. 2, pp. 112–122, Dec. 1973.

[42] A. Antonucci and D. Fontanelli, "Towards a predictive behavioural model for service robots in shared environments," in *Proc. IEEE Workshop Adv. Robot. Social Impacts (ARSO)*, Sep. 2018, pp. 9–14.

ALESSANDRO ANTONUCCI received the M.S. degree in mechatronic engineering from the University of Trento, Italy, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Information Engineering and Computer Science (DISI), Embedded Electronics and Computing Systems (EECS) Group. His research interests include human–robot interaction and pedestrian motion tracking and prediction.

GASTONE PIETRO ROSATI PAPINI graduated in automation engineering from the University of Pisa, in 2012. He received the Ph.D. degree from the Sant'Anna School of Advanced Studies in Emerging Digital Technologies, in 2016. He is currently an Assistant Professor of advanced control systems applied to the field of autonomous driving and ADAS with the Department of Industrial Engineering, University of Trento. He is the Co-Founder of Cheros s.r.l., a University of Pisa Spinout Company that deals with renewable energies and ICT solutions. He has over five years of experience in mechanics and control systems, recently focusing on the application of machine learning techniques for mechanic system modeling and control.



PAOLO BEVILACQUA received the master's degree in computer science and the Ph.D. degree in information and communication technology from the University of Trento, Trento, Italy, in 2015 and 2019, respectively. He is currently a Postdoctoral Researcher with the Embedded Electronics and Computing Systems (EECS) Group, Department of Information Engineering and Computer Science (DISI), University of Trento. His current research interests focus mainly on motion planning for wheeled robots, and on socially compliant autonomous navigation in environments shared with humans.

LUIGI PALOPOLI (Member, IEEE) received the Ph.D. degree from the "Scuola Superiore S. Anna, Pisa," in 2002. He is currently a Full Professor of computer engineering with the "Dipartimento di Ingegneria e Scienza dell'Informazione (DISI)," University of Trento. His research interests include real-time embedded control, formal methods, and stochastic analysis of real-time systems.

DANIELE FONTANELLI (Senior Member, IEEE) received the M.S. degree in information engineering and the Ph.D. degree in automation, robotics and bioengineering from the University of Pisa, Pisa, Italy, in 2001 and 2006, respectively. He was a Visiting Scientist with the Vision Laboratory, University of California at Los Angeles, Los Angeles, CA, USA, from 2006 to 2007. From 2007 to 2008, he was an Associate Researcher with the Interdepartmental Research Center "E. Piaggio," University of Pisa. From 2008 to 2013, he was an Associate Researcher with the Department of Information Engineering and Computer Science. Since 2014, he has been with the Department of Industrial Engineering, University of Trento, Trento, Italy, where he is currently an Associate Professor and he is leading the EIT-Digital international Master on "Autonomous Systems." He has authored or coauthored more than 150 scientific papers in peer-reviewed top journals and conference proceedings. He is also an Associate Editor-in-Chief of the IEEE Transactions on Instrumentation and Measurement and an Associate Editor of the IEEE Robotics and Automation Letters and the *IET Science, Measurement & Technology* journal. He has also served in the program committee for different conferences in the area of measurements and robotics. His research interests include distributed and real-time estimation and control, human motion models, localization algorithms, synchrophasor estimation, clock synchronization algorithms, resource aware control, wheeled mobile robots control, and service robotics.