H2020-ICT-2020-2 Grant agreement no: 101017274

DELIVERABLE 5.3

Representations and reasoning algorithms for HRSI

Dissemination Level: PUBLIC

Due date: month 48 (December 2024)

Deliverable type: Report

Lead beneficiary: UoL

Contents

1 Introduction

This deliverable (D) presents the results of the work carried out in Work Package (WP) 5, with a particular focus on Task (T) 5.3 and T5.4, which aim to enable efficient human-robot co-production (Objective O2). As part of WP5, the implementation of advanced solutions for human motion prediction and intention recognition was discussed in D5.1 and D5.2. In this deliverable, we introduce the other essential components of WP5 that contribute to achieving O2: novel context-aware representations and causal inference methods for human-robot spatial interaction (HRSI). T5.3 and T5.4 draw input from WP2 and WP3 regarding robot perception, semantic environment mapping, and localisation, and serve as input for WP6 and WP7 to generate safe and efficient robot motion plans.

The two tasks address the creation of context-aware representations for HRSI (T5.3) and the development of causal reasoning algorithms for safe HRSI (T5.4). In particular, T5.3 focuses on capturing the complexities of human spatial interactions and how they can influence robot motion in a possible production environment. By adopting a discrete motion representation based on a Qualitative Trajectory Calculus (QTC) [1, 2], we extend previous models to incorporate the relative motion of robots and humans, while accounting for both static and dynamic objects in the environment.

T5.4 aims to advance the understanding of HRSIs by learning causal models that account for the spatial behaviours of human agents and the possible contextual factors influencing their trajectories. By employing recent developments in causal inference, we develop algorithms tailored for robotic applications that efficiently learn causal models of HRSIs and compute intervention probabilities, facilitating safer and more efficient human-robot interactions. This task provide important high-level information for WP7 by reasoning on the HRSI causal models and estimating the risk associated with the robot's presence in particular areas of the environment.

The document is organized as follows: the activities carried out in T5.3 and T5.4 are presented in Section 2 and 3, respectively. Finally, Section 4 concludes the report summarising the main outcomes and discussing future developments.

2 Context-aware representations of HRSI

Human spatial interactions, defined as the mutual influence of motion behaviours between two or more people, depend on both human activities (e.g. speed and destination) and objects or constraints (e.g. nearby door, narrow corridor, etc). Similarly, HRSIs are influenced by the robot's motion (e.g. to approach the user) but also by other factors outside the direct control of the two interacting agents.

The scope of this task was to better capture the nature of HRSIs in realistic scenarios to enable safer and more socially acceptable robot motion behaviors. This was achieved by leveraging well-established representations of human-robot relative motion, particularly the 2D Qualitative Trajectory Calculus (QTC) [1, 3], to enhance the modelling and prediction of context-aware human motion and multi-agent spatial interactions.

QTC provides a discrete and symbolic motion representation, which addresses the challenges posed by continuous or purely quantitative descriptions in real-world environments. Building on its theoretical foundations and prior robotics applications [4, 2, 5], we refined previous QTC-based models to consider the influence of environmental and interaction-specific factors, including static objects (e.g., doors, pallets) and dynamic entities (e.g., humans). These models were initially studied using data from available datasets, but the ultimate aim was to create an interface capable of leveraging perception data from WP2 and robot localization inputs from WP3. By combining these inputs with

QTC-based motion representations, the task facilitates the prediction of HRSIs and provides a framework for enhancing the safety and social acceptability of robot behaviors in complex, dynamic environments.

2.1 Spatial Interactions: a Qualitative Formulation

A qualitative spatial interaction is defined by a vector of m QTC relations [1], which consist of qualitative symbols (q_i, $i \in \mathbb{Z}$) in the domain $U = \{-, 0, +\}$. We can distinguish between four types of QTC: 1) QTC_B basic, 2) QTC_C double-cross, 3) QTC_N network, and 4) QTC_S shape. Here, we focus on the use of QTC_C, since it better represents the dynamics of the agents in our application scenario. Two variations of QTC_C exist in the literature: QTC_{C_1}, with four symbols $\{q_1, q_2, q_3, q_4\}$, and QTC_{C_2}, with six symbols $\{q_1, q_2, q_3, q_4, q_5, q_6\}$. The symbols q_1 and q_2 represent the move towards/away (relative) motion between a pair of agents; q_3 and q_4 represent the left/right relation; q_5 indicates the relative speed, faster or slower; finally, q_6 depends on the (absolute) angle with respect to the reference line joining a pair of agents. Given the time series of two moving points, P_k and P_l, the qualitative interaction between them is expressed by the symbols q_i as follows:

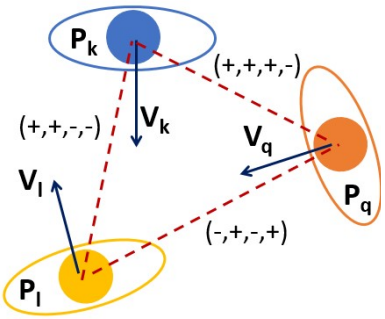


Figure 1: A case of QTC_{C_1} representation of interactions between three body points P_k, P_l, and P_q.

$(q_1) -: d(P_k|t^-, P_l|t) > d(P_k|t, P_l|t)$
$0: d(P_k|t^-, P_l|t) = d(P_k|t, P_l|t)$
$+: d(P_k|t^-, P_l|t) < d(P_k|t, P_l|t)$

(q_2) same as q_1, but swapping P_k and P_l

$(q_3) -: \|P_k^{t^+}\vec{P_k^t} \times P_l^{t}\vec{P_k^t}\| < 0$
$0: \|P_k^{t^+}\vec{P_k^t} \times P_l^{t}\vec{P_k^t}\| = 0$
$+:$ all other cases

(q_4) same as q_3, but swapping P_k and P_l

$(q_5) -: \|\vec{V_k^t}\| < \|\vec{V_l^t}\|$
$0: \|\vec{V_k^t}\| = \|\vec{V_l^t}\|$
$+:$ all other cases

$(q_6) -: \theta(\vec{V_k^t}, \vec{P_k P_l}^t) < \theta(\vec{V_l^t}, \vec{P_l P_k}^t)$
$0: \theta(\vec{V_k^t}, \vec{P_k P_l}^t) = \theta(\vec{V_l^t}, \vec{P_l P_k}^t)$
$+:$ all other cases.

where $d(.)$ is the euclidean distance between two points, $V(.)$ the velocity vector of a single point, $\theta(.)$ is the absolute angle between two vectors, and $\times$ is the cross-product operation. An example of QTC_{C_1} interaction is illustrated in Fig. 1 for three moving points.

2.2 Analysis of Qualitative Representations for Multi-Agent Spatial Interactions

In this part of the task we investigated the problem of Multi-Agent Spatial Interactions (MASI) in environments. To this end, we implemented three new neural network architectures applied to medium- and long-term interaction predictions, including different QTC-based representations. The resulting framework is explained below.

2.2.1 MASI Framework

Metrical motion information (i.e. plain coordinates and speed/orientation) of nearby agents helps robots navigate safely, but it might be not enough to reason about implicit

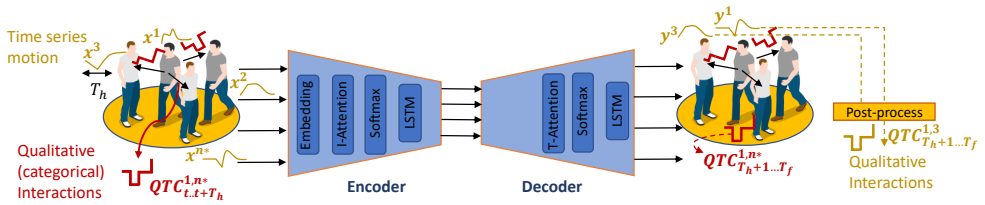


Figure 2: An input-temporal attention mechanism for predicting spatial interactions of multi-dimensional input categorical (red) and metrical (yellow) time series extracted from dense scenes: application to JackRabbot dataset. The diagram is extended from [6]. **x** is the input driving vector, **y** is the label vector.

Figure 3: JackRabbot dataset scenes: (top) *bytes-cafe-2019-02-07_0*, (bottom) *packard-poster-session-2019-03-20_2*.

intent conveyed through motion (e.g. a person's accelerating pace suggests some urgency). To address this, we proposed a Multi-Agent Spatial Interaction (MASI) framework that can be used to predict and compare *qualitative* interactions. Three variants of the framework (F) have been implemented:

1. F^{QTC-4}: symbol-driven approach to analyse and predict qualitative interactions using QTC_{C_1} (explained in Section 2.1);
2. F^{QTC-6}: symbol-driven approach to analyse and predict qualitative interactions using QTC_{C_2} (also in Section 2.1);
3. F^{ts}: data-driven approach using raw (metrical) trajectories as input, and predicting QTC-based interactions as output.

The main difference between the two symbol-driven frameworks is that F^{QTC-4} assigns a greater importance to the left/right and towards/away dichotomies, neglecting the relative velocity and angle embedded in F^{QTC-6}.

Network Architecture In order to implement F^{QTC-4} and F^{QTC-6}, we modified the original architecture of [6] to deal with time-series of categorical data, representing symbolic knowledge of the spatial interactions between pairs of agents. We also extended the prediction horizon to medium (i.e. 48 time steps, or 3.2s) and longer (i.e. 72 time steps, or 4.8s) time horizons. A schematic representation of the modified network is shown in Fig. 2. The input attention encoder consists of an input attention layer (I-Attention) which weighs n^* spatial interactions in a radial cluster. The encoder is then followed by a decoder with a temporal attention layer (T-Attention), capturing the temporal dependencies in multi-agent interactions. The network encodes n^* input series (denoted by **x**), each of length T_h, and decodes n^* output labels (denoted by **y**), each of length T_f, where T_f is the predictive time horizon and T_h is the time history used for the temporal attention purpose.

Data Processing In order to approach the problem of reasoning in socially crowded environments, we implemented a crowd clustering approach for local interactions prediction. We applied the radial clustering approach on the JackRabbot (JRDB[1]) open-source dataset (Fig. 3). Two different values for the radius were chosen inspired by [7], specifically $R_1 = 1.2m$ and $R_2 = 3.7m$. The raw data were further processed to extract QTC representations of a spatial interaction between each pair of agents, whose dictionary index is then used as ground truth output for F^{QTC-4} and F^{QTC-6} approaches. In parallel, the raw metric data are directly used as ground truth labels for the F^{ts} approach.

The environments considered in JRDB are fairly crowded. Among them, we selected a cafe shop (*bytes-cafe-2019-02-07_0*) for comparing the proposed prediction approaches, and two poster session scenarios (*packard-poster-session-2019-03-20_2*, denoted PS-2, and *packard-poster-session-2019-03-20_1*, denoted PS-1) for testing the framework on a domain-shift situation. For both the cafe and the poster sessions scenarios, we evaluated the prediction performance for a medium ($T_f = 3.2s$) and a longer term ($T_f = 4.8s$) horizons.

2.2.2 Experiments

The three proposed framework configurations implement the same architecture as in Fig. 2 but they were trained with different losses, since the input data is different. F^{QTC-4} and F^{QTC-6} were trained by minimising a categorical cross-entropy loss, while F^{ts} was trained using the root mean square error loss function (RMSE). Consequently, in order to compare their performance we use the so-called "conceptual QTC distance" [1] already defined in 1.

Testing Evaluation Table 1 shows the results for the three frameworks on the cafe scene, using cluster radius $R_1 = 1.2$m and $R_2 = 3.7$m.

On the test set, F^{QTC-6} significantly outperforms F^{QTC-4} over both medium and long time horizons. However, F^{ts} achieved the best performance across both horizons, with $F^{ts,1}$ (using QTC_{C_1}) excelling in medium-term predictions (3.2s) and $F^{ts,2}$ (using QTC_{C_2}) excelling in long-term predictions (4.8s). Considering the larger cluster radius, R_2, which accounts for more context, $F^{ts,1}$ outperforms all other configurations on both the medium and long horizons. It also outperforms $F^{ts,2}$ over $T_f = 4.8s$ and when R_1 is used. We can infer that with a larger cluster radius, more context is incorporated, which helps improve long-term predictions. As a result, fewer interaction symbols are needed to accurately represent the true interactions between multi-agent systems.

Domain-Shift (DS) Evaluation To further assess the generalisation capabilities of the three approaches, we re-trained and compared the results on another crowded environment (poster session PS-2, as shown in Fig. 3-bottom) with $R_1 = 1.2$m, and tested them on a different but related scenario (poster session PS-1). The performance on the testing set (i.e., 10% of PS-2) is reported in Table 2 (first column). We observe that $F^{ts,1}$ outperforms F^{QTC-4} and F^{QTC-6} in both medium- and long-term predictions. Notably, even within the same network configuration, $F^{ts,1}$ outperformed $F^{ts,2}$.

When looking at the transfer domain PS-1 in Table 2 (second column), all configurations succeeded in generalising to PS-1 on the medium and longer terms, except $F^{ts,1}$ and $F^{ts,2}$, which generalised well only on the medium term. Nevertheless, $F^{ts,1}$ continues to show the best performance overall when considering only PS-1.

In summary, we can conclude that $F^{ts,1}$ is the best framework for developing qualitative predictive solutions to embed a social autonomous system with additional intelligent

[1] https://jrdb.erc.monash.edu/

	Cafe			
	$\mu^{10\%-R_1}$	$\sigma^{10\%-R_1}$	$\mu^{10\%-R_2}$	$\sigma^{10\%-R_2}$
$\mathbf{F}^{QTC-6}$ (3.2s)	1.772	3.568	3.064	3.851
$\mathbf{F}^{QTC-4}$ (3.2s)	7.545	4.067	3	3.857
$\mathbf{F}^{ts,1}$ (3.2s)	**0.464**	**0.22**	**0.32**	**0.16**
$\mathbf{F}^{ts,2}$ (3.2s)	0.68	0.166	0.638	0.11
$\mathbf{F}^{QTC-6}$ (4.8s)	3.44	4.4	3.46	4
$\mathbf{F}^{QTC-4}$ (4.8s)	7.61	4.057	3.8	4.18
$\mathbf{F}^{ts,1}$ (4.8s)	3	1.254	**0.25**	**0.18**
$\mathbf{F}^{ts,2}$ (4.8s)	**0.644**	**0.146**	0.55	0.13

Table 1: Performance comparison between the QTC prediction approaches F^{QTC-4} and F^{QTC-6}, and the motion prediction-based QTC analysis framework F^{ts} evaluated on QTC_{C_1} ($F^{ts,1}$) and QTC_{C_2} ($F^{ts,2}$), in the cafe scene of JRDB and over $T_f = 3.2s$ and $4.8s$ prediction horizons. All measures are unitless. μ and σ are the normalised mean and standard deviation of the conceptual distance (d_{QTC}) measure over the test set. R_1 and R_2 correspond to cluster radius 1.2m and 3.7m, respectively. The best performance is highlighted in bold.

	PS-2		PS-1	
	$\mu^{10\%}$	$\sigma^{10\%}$	$\mu^{100\%}$	$\sigma^{100\%}$
F^{QTC-6}(3.2s)	1.78	3.558	0.77	0.26
F^{QTC-4}(3.2s)	7.34	4.08	1.3	1.86
$F^{ts,1}$(3.2s)	**0.49**	**0.217**	**0.43**	**0.22**
$F^{ts,2}$(3.2s)	0.715	0.17	0.7	0.17
F^{QTC-6}(4.8s)	2.098	3.81	0.84	0.26
F^{QTC-4}(4.8s)	8.018	3.7	1.27	1.92
$F^{ts,1}$(4.8s)	**0.297**	**0.21**	**0.35**	**0.22**
$F^{ts,2}$(4.8s)	0.547	0.137	0.6	0.15

Table 2: Performance comparison between the QTC prediction approaches F^{QTC-4} and F^{QTC-6}, and the motion prediction-based QTC analysis framework F^{ts} evaluated on QTC_{C_1} ($F^{ts,1}$) and QTC_{C_2} ($F^{ts,2}$), in the poster halls PS-2 and PS-1 of JRDB and over $T_f = 3.2s$ and $4.8s$ prediction horizons. All measures are unitless. μ and σ are the normalised mean and standard deviation of the conceptual distance (d_{QTC}) measure over 10% test set (PS-2) and 100% test set (PS-1). The best performance is highlighted in bold.

capabilities, such as inferring implicit intent communication and/or predicting needs from surrounding agents. $F^{ts,1}$ demonstrates the lowest mean and standard deviation loss over both short and long horizons and across different cluster radius.

In this activity, we presented and compared three network architectures for multi-agent analysis and prediction of qualitative interactions in dense social scenes, combining a symbolic motion representation with a dual-attention mechanism (input plus temporal). Specifically, we compared two symbol-driven neural networks for QTC prediction, F^{QTC-4} and F^{QTC-6}, with a metrical data-driven one, F^{ts}. We showed that the latter solution outperforms the previous two, suggesting that QTC alone is not sufficiently informative to capture the salient properties of human spatial interactions in complex social contexts.
This work was published in the paper titled "Qualitative Prediction of Multi-Agent Spatial Interactions", accepted at the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) [8].

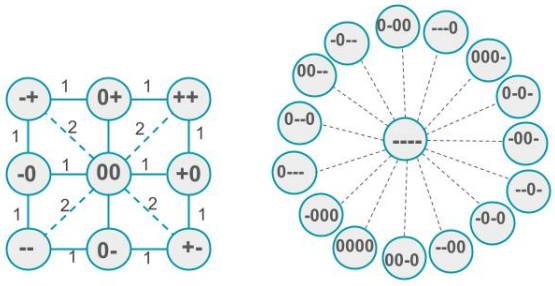


Figure 4: (left) The complete CND for QTC_{B1} in n-dimensional space. The QTC_{B1} has only q_1 and q_2 symbols. Straight edges represent a conceptual distance of 1, while dashed edges represent a conceptual distance of 2. (right) One part of the CND for QTC_{C1} illustrating the 15 possible transitions from the QTC state $\{-,-,-,-\}$, whose weight is therefore $\alpha_{cnd} = 0.067$.

2.3 A Neuro-Symbolic Approach for Enhanced Human Context Representation

In this task we aimed to advance context representation for reasoning on human motion in complex scenarios by addressing limitations in existing methods. Most prior works in this area embed all interactions equally within a pre-defined neighborhood size around a single agent, without distinguishing the varying relevance of interactions. In contrast, we formulated the problem of context-aware human motion modelling in terms of *weighted* interactions between pairs of agents. By incorporating a-priori information about the qualitative nature of spatial interactions, we demonstrate that it is possible to enrich the representation of human motion behaviours and improve their prediction accuracy.

To achieve this, we developed a neuro-symbolic model (NeuroSyM) that integrates a novel weighting mechanism for spatial interactions in the robot's environment. This mechanism leverages the probability of QTC transitions to label and weight such interactions and use them as symbolic inputs of a neural network architecture. The proposed approach was evaluated experimentally on medium- and long-term time horizons, using two state-of-the-art architectures, one for human motion prediction and the other for generic multivariate time-series prediction. Comprehensive testing was conducted on six challenging datasets of crowded scenarios, gathered from both fixed and mobile cameras. These experiments demonstrate the effectiveness of NeuroSyM and its potential to enhance the accuracy of context-aware human motion prediction in realistic environments.

2.3.1 QTC-based Interaction Weighting

Here we focus on QTC_{C_1}, explained in Section 2.1, to represent pairwise spatial interactions with the first four symbols, since these are the most informative and robust to noisy data. We label the latter exploiting the so-called Conceptual Neighborhood Diagram (CND) of QTC theory [9]. The CND is built on the notion of conceptual distance (**d**), which quantifies the closeness between QTC states at time t and t':

$$\mathbf{d}_{QTC^t}^{QTC^{t'}} = \sum_{q_i} | \, q_i^{QTC^t} - q_i^{QTC^{t'}} \, | \tag{1}$$

In Fig. 4 (left), each link (edge) between two QTC neighbours (nodes) includes the conceptual distance between the latter. In a CND, due to the laws of continuity [9], the conceptual neighbours of any particular QTC state are only a subset of all the possible states. For example, QTC_{C_1} can express 81 relations in total (each one of the four symbols q_i can assume 3 different values in U) but the conceptual neighbours of $\{-,-,-,-\}$ are

(a) Zara dataset.

(b) University dataset.

(c) JackRabbot cafe *bytes-cafe-2019-02-07_0* scene

Figure 5: Examples from UCY and JackRabbot datasets.

only 15, as illustrated in Fig. 4 (right). Note that the figure does not show the entire CND for QTC_{C_1} because it is too complex to visualise on a two-dimensional medium.

For each of the 81 states in QTC_{C1}, we assign a weight α_{cnd} formulated as follows:

$$\alpha_{cnd} = \Pr(QTC^{t'}|QTC^{t}) = \frac{1}{N_{Tr}} \tag{2}$$

where N_{Tr} represents the number of transitional states. In practice, α_{cnd} represents the level of stability, or reliability, of a QTC state. The higher the number of possible transitions from that state, the lower their likelihood to occur, and vice-versa. In

Given an interaction at time t, we associate a weight α_{cnd} to the interaction at $t+1$. In the literature, an interaction between agents A and B is usually represented by an embedding of their relative pose as follows:

$$Inter_{AB} = Dense(X_B - X_A) \tag{3}$$

where $Dense()$ is the embedding layer. The QTC state between A and B induces a specific α_{cnd} value, loaded from a dictionary of pre-computed weights, yielding a modified embedding $\alpha_{cnd}\ Inter_{AB}$. From a practical point of view, this symbolic knowledge of interactions between two moving points can be readily exploited by any neural architecture for context-aware motion prediction, since the CND dictionary, associating QTC states to their corresponding α_{cnd}, remains the same regardless of the data distribution.

2.3.2 Experiments

We evaluated our neuro-symbolic approach on a human motion prediction task, using two state-of-the-art architectures with raw trajectory data as input:

1. socially-acceptable trajectories with generative adversarial networks (SGAN) [10]: a well-known baseline for human motion prediction in crowded environments. It relies on:
    - ETH dataset [11] (sequences ETH and Hotel) — captured from fixed top-down cameras in public spaces.

- UCY dataset [12] (sequences Zara01-02 and Univ) — captured from a mobile stereo rig mounted on a car.

2. Dual-Stage Attention-Based Recurrent Neural Network (DA-RNN) [6]: a generic time-series forecasting network that supports the integration of dynamic and static contexts while paying attention to individual interactions.

To generalise our evaluation to robotics applications, we also used the JackRabbot (JRDB) dataset [13], which provides multi-sensor data of human behaviour from a mobile robot's perspective in indoor and outdoor environments. Unlike ETH and UCY, JackRabbot captures local interactions through on-board sensors such as 360° LiDAR (Velodyne) and fisheye cameras, making it particularly relevant for social robot navigation scenarios. To this end, we chose to use JackRabbot on a generic network architecture for time series prediction and where the following features can be incorporated: (a) the ability to integrate a dynamic context; (b) the ability to integrate key static objects of potential interactions (e.g. door, table, bar), differently from S-LSTM and SGAN; (c) the ability to test our neuro-symbolic approach on prediction architectures that, instead of using a pooling mechanism to overcome the size problem of dynamic input series (representing the neighbourhood in social scenarios), weights every single input (i.e. neighbour) by giving special attention to each one separately. One of the recent architectures that satisfy the last features is the dual-stage attention mechanism (DA-RNN) developed for time-series forecasting in [6].

Neuro-Symbolic SGAN – The core of SGAN [10] is a generator and a discriminator trained adversarially. The generator G produces candidate trajectories, while the discriminator model D estimates the probability that a sample comes from the training data (i.e. real) rather than from the generator output samples. The generator consists of an encoder and a decoder, separated by a pooling mechanism, while the discriminator is mainly an encoder. In SGAN, a variety loss is introduced on top of the adversarial (min-max) loss in order to encourage the generator to output diverse samples, thanks to a noise distribution injected to the pooling mechanism output. The performance measures used in SGAN for the evaluation process are the absolute displacement error (ADE) and the final displacement error (FDE) of the predicted trajectory ($\tilde{X}$). These are calculated as follows:

$$ADE = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} \|\tilde{X}_t^i - X_t^i\|_2}{N * T_{pred}} \qquad FDE = \frac{\sum_{i=1}^{N} \|\tilde{X}_{T_{pred}}^i - X_{T_{pred}}^i\|_2}{N} \tag{4}$$

where N is the total number of training trajectories.

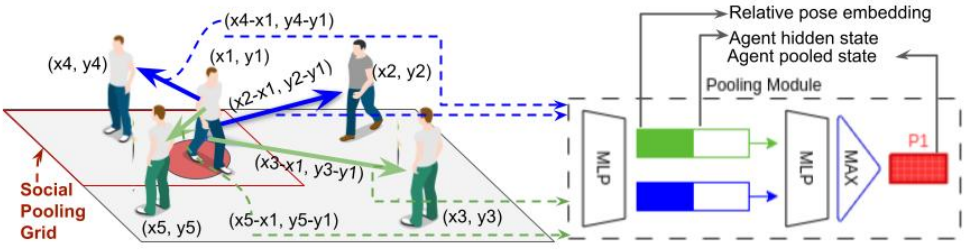


Figure 6: The neuro-symbolic SGAN pooling mechanism. The difference with the original SGAN pooling mechanism can be seen from the mixed arrows colour added within and outside the red grid to represent different types of spatial relations or interactions with the central agent standing on the red spot, which can be inferred from the NeuroSyM SGAN architecture.

Model	Measure	Baseline (SGAN)	NeuroSyM (SGAN)	Relative Gain (%)
Zara1	ADE	0.7 / 2.29	**0.21** / **0.34**	+70 / +85.15
	FDE	1.31 / 4.33	**0.41** / **0.7**	+68.7 / +83.8
	DE-STD	0.35 / 0.9	**0.22** / **0.4**	+37.14 / +55.5
	FDE-STD	1.1 / 2.86	**0.64** / **1.2**	+41.8 / +58
Zara2	ADE	0.44 / 0.95	**0.2** / **0.3**	+54.5 / +68.4
	FDE	0.84 / 1.85	**0.4** / **0.61**	+52.38 / +67
	DE-STD	0.26 / 0.52	**0.2** / **0.35**	+23 / +32.7
	FDE-STD	0.8 / 1.61	**0.58** / **1.05**	+27.5 / +34.78
Hotel	ADE	1.76 / 2.45	**0.35** / **0.5**	+80.10 / +79.6
	FDE	3.33 / 4.55	**0.67** / **0.99**	+79.88 / +78.2
	DE-STD	0.44 / 0.9	**0.32** / **0.56**	+27.27 / +37.7
	FDE-STD	1.51 / 2.88	**0.96** / **1.72**	+36.4 / +40.27
Univ	ADE	1.25 / 2.96	**0.36** / **0.62**	+71.2 / +79
	FDE	2.31 / 5.79	**0.74** / **1.31**	+67.9 / +77.37
	DE-STD	0.51 / 0.84	**0.24** / **0.38**	+52.9 / +54.7
	FDE-STD	1.6 / 2.63	**0.68** / **1.14**	+57.5 / +56.6
ETH	ADE	0.88 / 3.8	**0.63** / **0.73**	+28.4 / +80.78
	FDE	1.63 / 6.71	**1.25** / **1.44**	+23.3 / +78.5
	DE-STD	0.37 / 1.02	**0.37** / **0.64**	+0 / +37.2
	FDE-STD	1.15 / 3.29	**1.09** / **1.91**	+5.2 / +41.9
Mean Gain	ADE	—	—	+60.84 / +78.58
	FDE	—	—	+58.4 / +76.97
	DE-STD	—	—	+28 / +43.5
	FDE-STD	—	—	+33.68 / +46.3

Table 3: Performance comparison between the baseline architecture SGAN and its neuro-symbolic approach across all datasets. We report results in the format of 8/12 prediction time steps. ADE, FDE, DE-STD, and FDE-STD measures are in meters and in bold is highlighted the better measure among the two approaches. The lower error the better. The mean gain represents the mean of the relative gains over the 5 datasets, hence it only applies to the relative gain rows.

The proposed neuro-symbolic version of SGAN is illustrated in Fig. 6, highlighting the difference to the original pooling mechanism of SGAN [10]. NeuroSyM acts mainly on the pooling mechanism of the predictive models, where it represents human-human interactions by (a) embedding first their relative pose in all the observed states of each agent through a dense layer, then (b) weighing the embedded relative pose based on the CND-inspired label (α_{cnd}) associated to the interaction at a previous time step, and finally (c) max-pooling the weighted embedding across neighbours in the global scene. On the contrary, the original SGAN considers relative poses at the final observed state only, with no attention given to the reliability or stability level the interactions might have to help inferring future states of the agent under consideration.

Results – For a reliable comparison between SGAN and NeuroSyM SGAN, we trained again the former on our computing system (11th Gen Intel® Core™ i7-11800H processor and NVIDIA GeForce RTX 3080 16GB GPU), which was able to replicate almost the same hyper-parameters of the original work on SGAN, except for the batch size, in our case limited to 10 instead of 64. The ADE and FDE results for both architectures are reported together with their standard deviations (DE-STD and FDE-STD) in Table 3 for $T_{pred} = 8$ steps (i.e. 3.2 seconds) and 12 steps (i.e. 4.8 seconds), and on the five sequences from the publicly available datasets ETH and UCY. The results show a better ADE, FDE, DE-STD, and FDE-STD for the NeuroSyM approach compared to the original SGAN. The relative gain in terms of error drop is represented in Table 3 by a positive percentage for all the four measures with NeuroSyM with respect to SGAN on each dataset. The average relative gain for ADE, FDE, DE-STD, and FDE-STD, over the 5 datasets, is 60.84%, 58.4%, 28%, and 33.68%, respectively, for $T_{pred} = 8$; and 78.58%, 76.97%, 43.5%, 46.3%, respectively, for $T_{pred} = 12$.

Neuro-Symbolic DA-RNN – The original DA-RNN architecture [6] implements a dual-stage attention mechanism for time-series forecasting. The dual-stage network consists of an encoder with an input attention module weighing the n^* time-series data spatially, each of length T_h, where T_h is the observed time history. The encoder is then followed by a decoder with a temporal attention layer, capturing the temporal dependencies in the input series. The encoder and decoder are based on an LSTM recurrent neural network. The network outputs the prediction of one time-series data of length T_f, where T_f is the predictive time horizon.

The proposed NeuroSyM version of DA-RNN leverages symbolic knowledge of the spatial interactions between pairs of agents. In DA-RNN, the encoder attention weights ("α" in Fig. 7) highlights the importance of each input series at time t on the output prediction at $t+1$. The input attention weights in DA-RNN are calculated as follows:

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^{n} \exp(e_t^i)} \tag{5}$$

where e_t^k is the embedding of the k^{th} input series at time t. It is implemented as:

$$e_t^k = dense[tanh(dense(\mathbf{h}_{t-1}; \mathbf{s}_{t-1}) + dense(\mathbf{x}^k_{1..T_h}))] \tag{6}$$

where h_{t-1} and s_{t-1} are the hidden and cell state of the encoder LSTM at a previous time step. The NeuroSyM DA-RNN acts on the input series embedding e_t^k before the softmax function (Eq. 5) is applied on it. Hence, the NeuroSyM approach transforms Eq. 6 into $\alpha^k_{cnd,t}\ e_t^k$, updating the encoder attention weights with an a-priori knowledge of the reliability or stability of each input series. For applications of human motion prediction in crowds (i.e. with context), $\alpha^k_{cnd,t}$ is generated from Eq. 2. Each input series represents the motion history of a neighbour agent, whereas the first time series is the motion history of the considered person and the output is the predicted motion of that specific agent. Fig. 7 illustrates schematically where the NeuroSyM module intervenes on the original DA-RNN architecture with the injection of a CND layer at the interface between the embedding and the softmax layers.

Data Processing – Crowded scenarios, such as those in the JackRabbot dataset, often involve an unpredictable number of individuals entering (P_e) and leaving (P_l) the environment, which can lead to a combinatorial explosion of input data points and training parameters. To address this, we implemented a crowd clustering approach for embedding local interactions:

- For each agent i, a cluster was generated with a fixed interaction radius of $R = 3.7$ m, based on proxemics literature [7].
- Each cluster includes n input series, representing the agents within the radius over a time interval T. To ensure computational feasibility, the maximum number of input series (n^*) was fixed, and smaller clusters were padded with complementary "fake" values.

The raw data from the JackRabbot dataset, consisting of annotated 3D point clouds, was processed to extract QTC representations of spatial interactions between agents. These representations were assigned weights ($\alpha^k_{cnd,t}$) using a Conceptual Neighborhood Diagram (CND) dictionary. These weights served as a priori information for the NeuroSyM architecture, enhancing the embedding of input data by reflecting the reliability of each input series. In addition to dynamic contexts, static context features such as bar order and check-out points, exit doors, and drinking water stations were manually identified and incorporated as input series. This approach was tested in a crowded cafe scenario (dataset *bytes-cafe-2019-02-07_0*).

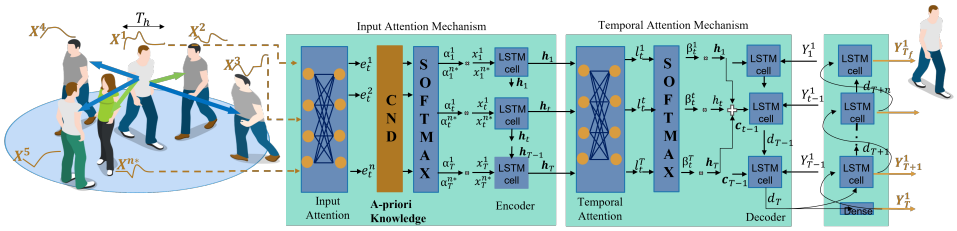


Figure 7: NeuroSyM approach for attention-based time-series analysis, extended from [6] and adapted to multi-step human motion prediction in crowded environments. Inputs are n^* time series centered on the cluster's primary agent, while outputs predict its trajectory. The CND layer adds a-priori knowledge ($\alpha^k_{cnd,t}$) to embeddings (**e**) and weights from QTC spatial relations of neighbors. Temporal attention weights (**l**) and context (**c**) are used in the encoder-decoder structure, with dense layers for input and attention mechanisms.

Architecture	RMSE	MAE
DA-RNN (Baseline)	3.61 / **3.572**	**2.097** / 2.753
NeuroSyM DA-RNN	**2.815** / 3.728	2.162 / **2.166**
Relative Gain (%)	+22 / -4.37	-3.1 / +21.32

Table 4: Performance comparison between the baseline architecture DA-RNN and the NeuroSyM approach on the JackRabbot dataset. The results' format refers to the 48/80 prediction time steps. RMSE and MAE values are in meters, and the best results are highlighted in bold (i.e. the lower error, the better).

Results – The performance of NeuroSyM DA-RNN was evaluated against the original DA-RNN for medium-term (48 steps, 3.2 s) and long-term (80 steps, 5.33 s) prediction horizons. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics were used to measure prediction accuracy on the JackRabbot dataset (Table 4).

We can clearly see that the NeuroSyM version of the architecture succeeds in decreasing the RMSE metric by 22% on the 48 steps prediction horizon, while influencing the performance negatively by 4% on the longer 80 steps horizon. At the same time, the neuro-symbolic approach decreased the MAE by 21% on the 80 steps horizon, while influencing negatively by 3% the 48 time steps prediction.

In this activity, we presented a neuro-symbolic approach for context-aware human motion representation (NeuroSyM) in dense scenarios. The approach leverages a qualitative representation of interactions between dynamic agents to assess and weight the influence of neighborhood interactions. The NeuroSyM approach was evaluated on two baseline prediction architectures, SGAN and DA-RNN. Our results demonstrated that NeuroSyM outperformed both architectures in most cases, particularly for medium- and long-term prediction horizons. These findings validate the effectiveness of integrating symbolic knowledge into neural models for enhancing human motion representation and reasoning.

The scientific results of this activity were published in the paper titled "A Neuro-Symbolic Approach for Enhanced Human Motion Prediction", accepted at the International Joint Conference on Neural Networks (IJCNN) [14]. A Python implementation of the NeuroSyM framework was also made available online[a].

[a] https://github.com/sariahmghames/NeuroSyM-prediction

2.4 Deployment and Evaluation of a ROS-based Neuro-Symbolic Model for Human Motion Representation

In the activities presented in Section 2.3, we proposed and evaluated NeuroSyM, a neuro-symbolic approach for context-aware human motion representation and prediction in dense scenarios, which integrates QTC-based information to assess and weight the influence of neighborhood interactions. However, the original method was not suitable yet for on-board and real-time applications, which limited its use in robotics scenarios.

The system presented in this section, called *neu***ROS***ym*, was developed to address those limitation by enabling the real-time execution of NeuroSyM using a concurrent stream of sensor data. Furthermore, being integrated in ROS, the resulting system can be directly deployed on a robot platform.

2.4.1 neuROSym Architecture

The new ROS package *neuROSym*, shown in Fig. 8, consists of the following three nodes:

- **Inference model node**: It implements two subscribers to the same observational data topic whose messages are generated by a human tracker library. In parallel, it implements two publishers for the data visualisation and analytics node. Each pair subscriber-publisher corresponds to either ground truth or predicted samples. The node implements also the inference model for the prediction method under investigation.
- **Data visualisation and analytics node**: This node runs in parallel to the inference node in order to generate, online, plots of the ground truth and predicted trajectories. It also generates average performance metrics, simultaneously to the visual plots.
- **Data post-processing node**: This node is required to perform corrections in case the human tracking system misses some detections.

The inference model node is based on our previous work [14], explained in Section 2.3, and incorporates both the NeuroSyM and the SGAN architectures.

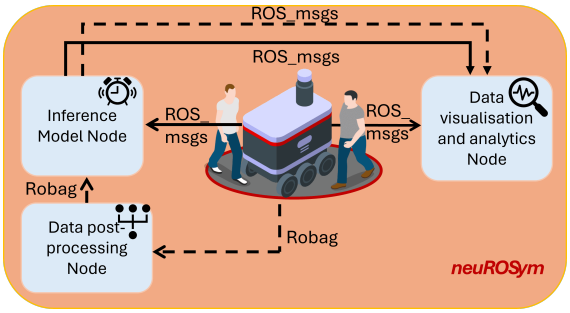


Figure 8: Deployment of *neuROSym* for online and context-aware human motion analysis, with real-time visualisation. The three blocks are ROS nodes, while the filled arrows and the dashed ones represent the online and offline inference, respectively. Each arrows label indicates the type of messages published and/or subscribed to by each node.

(a) (b) (c)

Figure 9: (a) Experimental Scenario A with two humans walking parallel to each other towards their goal (room end) and back, repetitively. The online trajectory prediction is performed by models trained on the UCY-Zara01 dataset. (b) Experimental Scenario B with two humans crossing each other's path. Here the models are trained on the THOR dataset. (c) RViz visualization of the Bayes People Tracker with human bounding-boxes extracted from the robot's LiDAR point-clouds.

2.4.2 Experiments

Experimental Setup – We used a TIAGo[2] mobile robot to monitor the motion of two people over a time period of 2 minutes. The robot was positioned at the corner of the experimental room (5m × 8.2m) and was equipped with a Velodyne VLP-16 3D LiDAR sensor, as shown in Fig. 9a. To track people in the scene, we run a Bayes People Tracker[3] [15] using point-cloud data from the LiDAR at a frequency of 10Hz. Fig. 9c shows an RViz screenshot with two humans tracked by the robot.

We conducted two types of experiments, illustrated in Fig. 9a and 9b. During these, we recorded the runtime of each inference model (SGAN baseline and NeuroSyM). We registered the rosbag file of each experiment (four in total) for offline evaluation.
Scenario A: "all-forward" motion behaviour. Both SGAN and NeuroSyM were trained on the UCY-Zara01 pedestrians dataset [12]. The motion pattern of the pedestrians resembles the *all-forward* motion pattern replicated in our experiments (i.e. people walking in parallel directions) and illustrated in Fig. 9a.
Scenario B: "cross-path" motion behaviour. The inference models were trained on the THOR dataset [16] and similar motion patterns were replicated in our *cross-path* scenario, as illustrated in Fig. 9b and 10.

Data Processing – The ROS inference node processes the data sequentially, with an observed time window of 8 samples. Human trajectories affected by tracking errors (e.g. because of occlusions) were filtered out and not considered. The ROS inference node and the visualisation node run simultaneously, showing predicted and ground-truth trajectories at runtime. The performance comparison between the baseline SGAN model and the NeuroSyM architecture was conducted on the recorded rosbag files.

Results and Discussion – We evaluated the average accuracy and runtime of each inference model over the 2-minutes sessions of both experimental settings. The results are reported in Table 5, which include average displacement error (ADE), final displacement

[2]https://pal-robotics.com/robots/tiago/
[3]https://github.com/LCAS/bayestracking

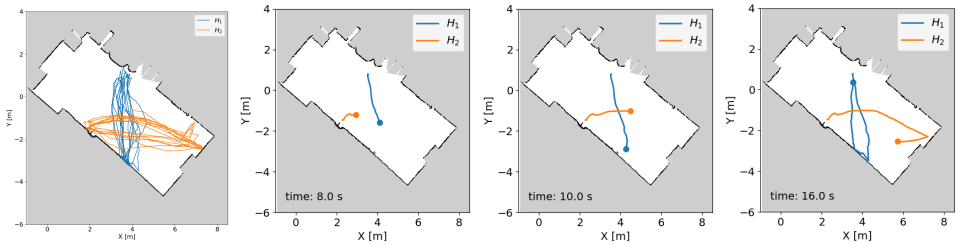


Figure 10: (Left) Full human motion trajectories in Scenario B, where two people H_1 and H_2 (dynamic objects) move back and forth to their destinations (static objects), crossing each other's path and avoiding collisions. Snapshots at frames t = 8 (centre-left), 10 (centre-right), and 16s (right).

Scenario		SGAN		NeuroSyM	
		rosbag1	**rosbag2**	**rosbag1**	**rosbag2**
A	**Avg. ADE** (m)	12.4	16.32	**7.06**	**2.52**
	Avg. FDE (m)	2.28	3.24	**1.31**	**0.68**
	Avg time (s)	**4.17**		**5.37**	
B	**Avg. ADE** (m)	10.88	24.27	**5.7**	**9.87**
	Avg. FDE (m)	2.67	5	**1.4**	**1.8**
	Avg time (s)	**5.19**		**7.36**	

Table 5: Accuracy and runtime evaluation for Scenario A and B, in terms of average displacement error (ADE), final displacement error (FDE) and time, over 2-minutes long experiments.

error (FDE) and inference time. These tables present 8 results in total, 4 for each scenario (A and B). We can see that, in all of the four cases, the higher accuracy achieved by NeuroSyM significantly reduced both ADE and FDE values compared to the SGAN baseline. We also evaluated the average runtime of each inference model in both experimental scenarios. Table 5, shows that NeuroSyM model is slightly slower than, but still comparable to, the SGAN baseline.

From Table 5, we can conclude that, although the NeuroSyM architecture requires more time to predict human trajectories compared to the SGAN baseline, it is still relatively fast and, with some code optimisation, suitable for real-time deployment. In particular, the trade-off between runtime and accuracy is clearly in favour of the NeuroSyM solution, since its QTC-based context-awareness enables more accurate motion predictions.

In this activity, we presented and tested neuROSym, a ROS package for neuro-symbolic human motion analysis and prediction with real-time visualisation. This package enabled the on-board implementation and evaluation of two inference models, SGAN and NeuroSyM, resulted from the activity in Section 2.3.
This work was presented in the paper titled "neuROSym: Deployment and Evaluation of a ROS-based Neuro-Symbolic Model for Human Motion Prediction", accepted at the IEEE Conference on Robotics, Automation and Mechatronics (RAM). [17].
Moreover, the software was made available on a public repository[a] and it was integrated on the DARKO platform.

[a] https://github.com/sariahmghames/neuROSym

3 Causal Reasoning for Safe HRSI

In the DARKO scenario, the robot operates in complex intralogistics settings, navigating dynamically within spaces shared by human workers. The DARKO robot must perform tasks efficiently and autonomously while anticipating and responding to human behaviour. It must complete its tasks with the awareness that its actions may trigger unpredictable responses from individuals nearby. Understanding the cause-effect relationships in the environment enables the robot to reason about its actions, enhancing both task execution and safety in human-robot collaboration.

The scope of this task was to define a framework that takes HRSI data from WP2 (perception) and WP3 (mapping and localisation) as input, reconstructs a causal representation between the features used to describe the interaction, and then reasons on the reconstructed causal model to enhance the interaction. The various steps involved in creating the framework are presented and discussed in the remainder of this section.

3.1 Causal Discovery of Dynamic Models for Human Spatial Interactions

This work aimed to reconstruct causal models to represent humans-goal, human-human and human-robot 2D spatial interactions, in single and multi-agent scenarios. To do this, a state-of-the-art causal discovery method has been exploited in a robotic application using time-series data from real-world sensors. The utility of the causal models has been assessed by predicting spatial interactions in human environments.

3.1.1 Causal Discovery from Observational data

The developed approach (depicted in Fig. 11) is based on the observation of human spatial behaviours to recover the underlying causal model. This causal analysis was performed by using the Peter & Clark Momentary Conditional Independence (PCMCI) causal discovery algorithm [18]. First, we identified some important factors (i.e. variables) affecting human motion in the considered scenarios, and from that we reconstructed the most likely causal links from real sensor data. Finally, we used the discovered causal models to forecast the latter with a state-of-the-art Gaussian Process Regression (GPR) technique [19], showing that the causality-based GPR improves the accuracy of the human interaction prediction compared to a non-causal version. Two different scenarios have been modelled and analysed.

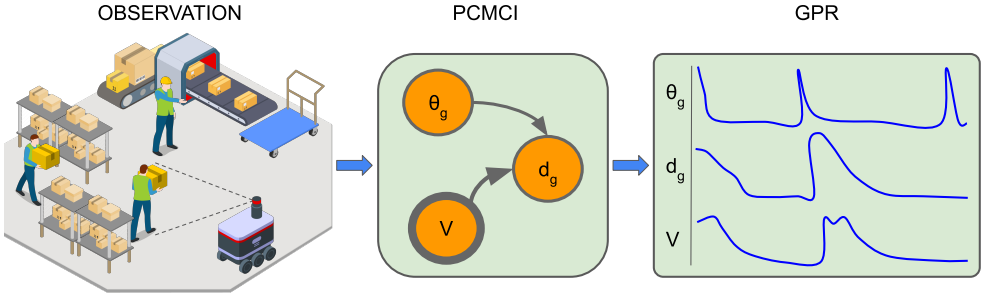


Figure 11: Causal prediction approach: a robot reconstructing a causal model from observation of human behaviours in a warehouse environment. The causal model is then used for human spatial behaviour prediction.

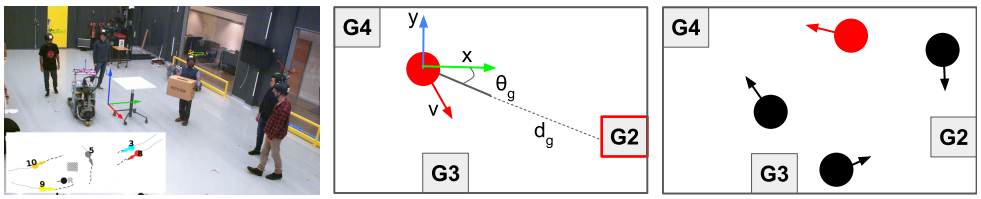


Figure 12: Image from THÖR dataset [16] (left). Representation of the two analysed scenarios: (centre) the human-goal scenario, (right) the human-moving obstacles scenario. The agents consist of a circle and an arrow specifying, respectively, the current position and orientation. The selected agent is red, while, the obstacles are black.

Human-goal scenario – The first scenario includes interactions between human and (static) goals in a warehouse-like environment, illustrated in Fig. 12 (centre), where the agent walks among different positions (grey squares) to move some boxes or grab/use some tools. The grey line connecting agent and goal specifies the angle θ_g between the two. The following features were deemed essential to explain the human motion behaviour: *(i)* angle agent-goal θ_g; *(ii)* euclidean distance agent-goal d_g; *(iii)* agent velocity v. The angle θ_g represents the human intention to reach a desired position (the person will first point towards the desired target before reaching it); then the person walks towards the goal, reducing the distance from it, at first by increasing the walking speed and finally decreasing, when close to the destination. Soon after the human has reached the goal, θ_g changes to the next one, and the process restarts. What we expect from this scenario are therefore the following causal relations:

(a) θ_g depends on the distance, when the latter decreases to zero then θ_g changes;
(b) d_g is inversely related to v and depends on θ_g;
(c) v is a direct function of the distance d_g.

Human-moving obstacles scenario – The second scenario involves multiple agents. It reproduces the interaction between a selected human and nearby dynamic obstacles (e.g. other humans, mobile robot), as shown in Fig. 12 (right). In this case, we take into account human reactions to possible collisions with obstacles, modelled by a *risk* factor. Consequently, the relevant features in this scenario are *(i)* euclidean distance d_g of the selected agent-goal, *(ii)* agent's velocity v, and *(iii)* *risk* value. The agent moves between goals in the environment, so the cause-effect relation between distance and velocity will be similar to the previous scenario. The main difference in this case is that, instead of reaching the goal without problems, the agent needs to consider the presence of other obstacles, and the interactions with them will affect the resulting behaviour. In particular, the agent's velocity is affected by possible collisions (e.g. sudden stop or direction change to avoid an obstacle). The expected causal links in this scenario are the following ones:

(a) d_g depends inversely on v;
(b) v is a direct function of the d_g, but it is also affected by the collision $risk$;
(c) $risk$ depends on the velocity, as explained below.

In order to model a numerical $risk$ value as a function of the agent's interactions, we implemented a popular strategy named Velocity Obstacles (VO) [20]. The VO technique identifies an unsafe sub-set of velocities for the selected agent that would lead to a collision with a moving or static obstacle, assuming the latter maintains a constant velocity. The risk can then be defined as follows. At each time step, we apply the VO to the agent's closest obstacle. Such risk is a function of two parameters, both depending on the selected agent's velocity (i.e. point P inside the VO; see Fig. 13):

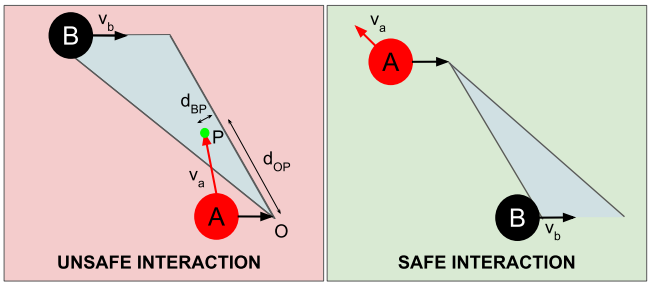


Figure 13: Velocity Obstacle (VO) technique. A Collision Cone (CC) is built from the selected agent A to the enlarged encumbrance of the obstacle B. Then, the CC is translated by v_b to identify the VO, which partitions the velocity space of A into *avoiding* and *colliding* regions, i.e. velocities lying outside and inside the VO, respectively. (Left) an interaction leading to a collision. (Right) a collision-free interaction.

- d_{OP}, the distance between the cone's origin O and P, which is proportional to the time available for the selected agent A to avoid the collision with B;
- d_{BP}, the distance between P and the closest cone's boundary, which indicates the steering effort required by A to avoid the collision with B.

Consequently, the risk of collision is defined as follows:

$$risk = e^{d_{OP}+d_{BP}+v_a}. \tag{7}$$

In order to avoid mostly-constant values (undetectable by the causal discovery algorithm), we introduced a third parameter v_a, which is the velocity of the selected agent.

Causal prediction with PCMCI and GPR – Our approach for modeling and predicting spatial interactions, shown in Fig. 11, can be decomposed in three main steps: (*i*) extract the necessary time-series of sensor data from the two previously explained scenarios; (*ii*) use them for the causal discovery performed by the PCMCI algorithm; (*iii*) finally, embed the causal models in a GPR-based prediction system. More details about the PCMCI causal discovery method in [18]. In the last step, we exploit the GPR, a nonparametric kernel-based probabilistic model [19], to build a causal GPR predictor, useful to forecast each variable by using only its parents, and not all the variables involved in the scenario, as a non-causal GPR predictor would do.

3.1.2 Experiments

We evaluated our approach for causal modeling and prediction of human spatial behaviours on two challenging datasets: THÖR [16] and ATC Pedestrian Tracking [21]. Both contains data of people moving in indoor environments, a workshop/warehouse and a shopping center, respectively. Our strategy was first to extract the necessary time-series from the two datasets, and then use it for causal discovery. In order to prove the usefulness of the causal models, a comparison between causal and a non-causal predictions was finally conducted. We considered two different datasets in order to verify, for the human-goal scenario that the discovered causal model holds for similar human behaviours, even when observed in different environments. The human-moving obstacle scenario, instead, was used to demonstrate that it is possible to perform causal discovery for other types of human spatial interactions (i.e. with collision avoidance).

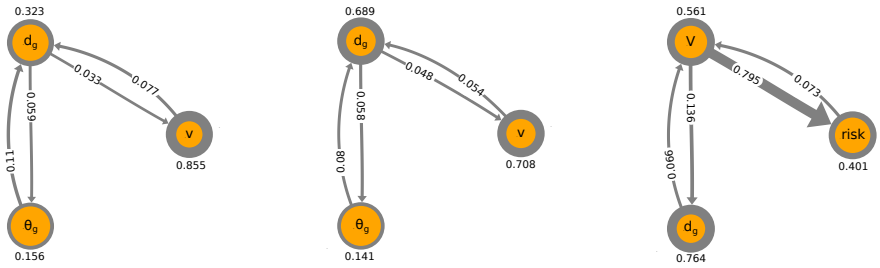


Figure 14: Causal models of: human-goal scenario with (left) THÖR and (centre) ATC datasets, human-moving obstacles scenario with (right) THÖR dataset. The thickness of the arrows and of the nodes' border represents, respectively, the strength of the cross and auto-causal dependency, specified by the number on each node/link (the stronger the dependency, the thicker the line). All the relations correspond to a 1-step lag time.

Data processing – From both datasets, we extracted the x-y positions of each agent and derived all the necessary quantities from them (i.e. orientation θ, velocity v, etc.).

- **THÖR dataset:** it provides a wide variety of interactions between humans, robot, and static objects (Fig. 12, left). We used this dataset to analyse both human-goal and human-moving obstacles scenarios.
- **ATC pedestrian tracking dataset:** the data was collected in the large atrium of a shopping mall (much bigger than THÖR's environment). Due to its large area and crowd, this dataset was not suitable for the collision-enhanced scenario. Therefore, we used this dataset only for the human-goal scenario.

Results – We applied PCMCI for causal discovery, using Gaussian Process Regression and Distance Correlation (GPDC) [22], with a 1-step lag time, where variables at time t could only be influenced by those at time $t-1$. The resulting causal models are shown in Fig. 14, with the thickness of arrows representing the strength of causal dependencies. These models support the hypothesis that:

- The human-goal scenario causal model generalises across datasets (THÖR and ATC), with causal strengths varying due to different sampling frequencies and noise levels.
- The human-moving obstacle scenario produced a different causal model, highlighting the ability to handle varied human behaviours.

We used the discovered causal models to predict key spatial interaction variables in both scenarios, such as orientation (θ_g), distance (d_g), and velocity (v). To evaluate the utility of the causal models, we compared a causally-informed Gaussian Process Regression (GPR) model with a non-causal GPR model. The comparison was made using the Normalised Mean Absolute Error (NMAE) metric, definition of which can be found in [23].

Fig. 15 shows that the causal model consistently improved prediction accuracy, particularly for variables like θ_g and v in the human-goal scenario using the THÖR dataset. For the distance variable (d_g), both models yielded similar performance, as the full set of predictors was used in both causal and non-causal cases. The NMAE comparison, summarised in the bar chart (bottom-right), demonstrates that the causal model led to more accurate predictions. Table 6 shows the mean NMAE for all considered scenarios. The causal GPR approach consistently outperformed the non-causal model in all cases. The ATC dataset, with its longer time-series, had a higher mean NMAE for the human-goal scenario, likely due to differences in dataset characteristics. Further details and discussion of the results can be found in our paper [23].

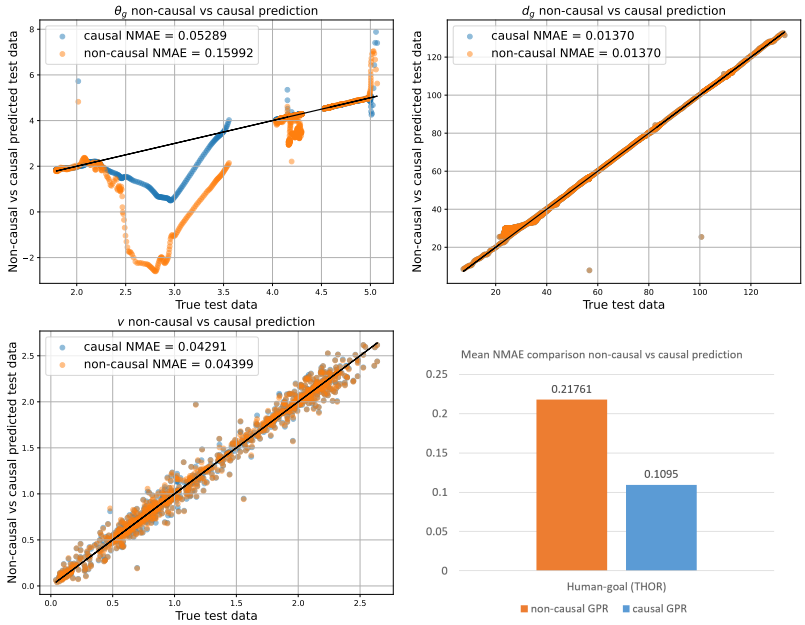


Figure 15: Comparison between non-causal and causal GPR prediction and NMAE in the human-goal scenario of the THÖR dataset for the spatial interaction variables θ_g (top-left), d_g (top-right), and v (bottom-left). A bar chart (bottom-right) summarises the comparison using the mean NMAE over all three variables.

	Human-goal		Human-moving obs
	THÖR	ATC	THÖR
Non-causal	0.21761	1.61692	0.37849
Causal	**0.1095**	**1.54552**	**0.36453**

Table 6: Mean NMAE of causal and non-causal predictions over the involved variables for both scenarios and datasets.

This activity demonstrated the feasibility of using state-of-the-art causal discovery methods, specifically the PCMCI algorithm, to reconstruct causal models of Human-Robot Spatial Interactions (HRSI). We successfully applied this method to recreate causal models for two different HRSI scenarios, showing that these models are valuable for predicting HRSI.

The findings from this activity have been consolidated into the paper titled "Causal Discovery of Dynamic Models for Predicting Human Spatial Interactions", presented at the International Conference on Social Robotics (ICSR) [23].

3.2 Enhancing Causal Discovery from Robot Sensor Data in Dynamic Scenarios

From the work presented in Section 3.1, we discovered that it is indeed feasible to reconstruct causal models of HRSI using the PCMCI causal discovery method. However, a significant challenge emerged: causal analysis of complex and dynamic systems is extremely demanding in terms of both time and hardware resources, as also noted in [24, 25]. This poses a challenge for autonomous robotics, which often operate with limited hardware resources and real-time constraints.

The primary factor contributing to PCMCI's computational cost is the *number of variables* involved in the causal analysis. In this part of the deliverable, we describe the activities conducted to extend one of the state-of-the-art causal discovery methods, PCMCI [18], by augmenting it with a feature-selection algorithm capable of identifying the correct subset of variables to include in the causal analysis from a predefined set. Consequently, we introduced an all-in-one approach that identifies the causal features representing the system and uses them to build a causal model directly from time-series data. This modification makes the causal discovery process faster and more accurate.

For instance, in an DARKO-inspired automated warehouse scenario (see Fig. 16), where a robot observes the interactions among objects and humans (e.g. worker and shelf), it is important to know which features, among those detectable by the robot's on-board sensors, are relevant for describing the observed interaction (e.g. human-shelf distance/angle, human velocity, etc.), and which instead can be neglected (e.g. other humans not involved in the interaction). The approach, proposed in this section, allows the robot to discard unnecessary features and build a causal model of the interaction using only those actually involved in the process.

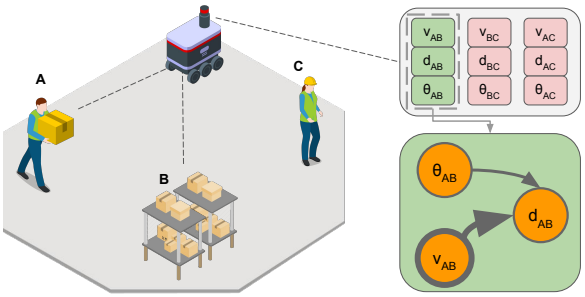


Figure 16: A mobile robot in a warehouse-like environment observes the interaction between agents A and B. By using our approach, the robot can disregard the interactions AC and BC as agent B is a static object and agent C is a standing human, not involved in the interaction.

3.2.1 Filtered-based Causal Discovery

To reduce the computational cost of PCMCI and enable its execution onboard the robot, we enhanced it by introducing a Transfer Entropy (TE)-based feature selection method. TE is an extension of mutual information that quantifies the directed information transfer between the time-series of a source and a target variable and it can effectively indicate whether a relationship between two variables exists. Further details regarding the limitations and challenges of using TE as a causal measure are discussed in the paper resulting from this activity [26].

The developed approach, named Filtered PCMCI (F-PCMCI), uses a TE-based method to "filter" the important features and their possible associations from the whole set of variables, before the actual causal analysis. A Python implementation of F-PCMCI has been developed and made publicly available[4]. We used TE to decide which variables and links can be excluded from the original set, and those which are needed for the causal analysis. As output, the filter returns a set of variables and a hypothetical causal model, which then needs to be validated by a proper causal analysis. The latter is performed by the PCMCI causal discovery algorithm. A pseudo-code implementation and a block diagram of our approach are also illustrated in Algorithm 1 and Fig. 17, respectively.

3.2.2 Experiments

To evaluate our approach and verify its advantages in terms of computational cost and causal models' accuracy with respect to PCMCI, we first validated it with the toy problems

[4]https://github.com/lcastri/fpcmci

Algorithm 1 F-PCMCI

Require: time-series data D, significance threshold α, min and max time lag τ_{min}, τ_{max}

$CS = \{\}$ ← hypothetical causal structure dictionary
for each target T in D **do**
  $S_T = \emptyset$ ← T sources / conditioning set
  $L = [\ ]$ ← temporary list
  while D not empty **do**
    for each source S in $D \setminus T$ **do**
      $(p\text{-}value, I)_S = \mathrm{TE}_{S \to T|S_T}(\tau_{min}, \tau_{max})$
      add $(p\text{-}value, I)_S$ to L
    $(p\text{-}value, I)_{S_b} = \arg\max_I(L)$ ← best candidate
    if $p\text{-}value \leq \alpha$ **then**
      remove S from D and add S to S_T
    else
      if $S_T \neq \emptyset$ **then** $CS(T) = S_T$
      break
$D_s \leftarrow$ shrink original D by var_{sel} =`keys`(CS)
CM = `PCMCI`(D_s, α, τ_{min}, τ_{max}, CS)
return CM ← causal model

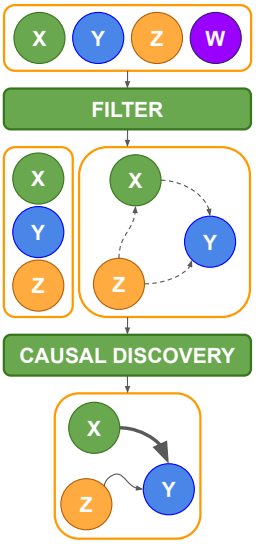


Figure 17: F-PCMCI block-scheme representation with an example.

and with Functional Magnetic Resonance Imaging (fMRI) time-series data generated with a tool[5] provided by [27]. The latter is able to generate realistic and rich simulations of fMRI time-series data with ground-truth brain networks. The evaluation strategies supported our hypothesis that F-PCMCI offers advantages in terms of both computational cost and accuracy compared to PCMCI. The results and discussion of the evaluation strategies can be found in the paper resulting from this activity [26].

Once, established that our approach works correctly, we used it for modeling and predicting human spatial behaviours on a challenging dataset, i.e. THÖR [16]. Our strategy is first to extract the real sensor time-series data from the dataset, as already explained in Section 3.1.2, and then use them for causal discovery. The effectiveness of our approach is demonstrated by comparing causal and non-causal predictions. A further comparison between PCMCI and F-PCMCI is provided to illustrate the advantages of our method with respect to the state-of-the-art.

Modeling and Predicting Real-world Human Spatial Interactions – Finally, we applied our approach to model and predict spatial interactions (Fig. 12, left). This application involves three main steps:

1. extracting time-series of sensor data from human spatial interaction scenarios;
2. reconstruct the causal model using F-PCMCI;
3. embedding the causal model in a LSTM-based prediction system.

To extract time-series data from human spatial interaction scenarios (Step 1), we utilized the THÖR dataset, specifically extracting the x-y positions of each agent, as described in Section 3.1.2. From these positions, we derived additional quantities necessary for the analysis, as detailed below. To represent human spatial interactions, we identified 8 variables for each agent that are suitable for this application. These variables were then used in the subsequent causal analysis.

[5] https://www.fmrib.ox.ac.uk/datasets/netsim/

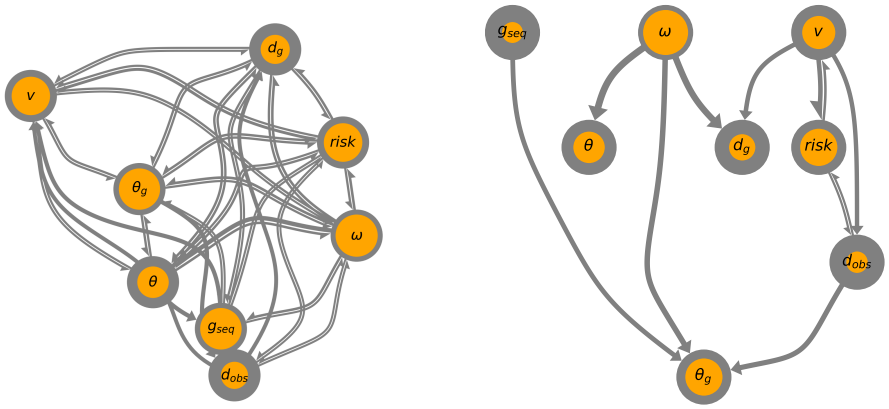


Figure 18: Causal models of the THÖR dataset using PCMCI (left) and F-PCMCI (right). Arrows and borders of the nodes represent the strength of cross-causal and auto-causal dependencies, with stronger dependencies shown by thicker lines/borders. All dependencies have a 1-step lag time.

1) d_g – distance between the current position of the agent and its goal;
2) v – velocity of the selected agent;
3) $risk$ – risk of collision with other agents (as explained in Section 3.1.1);
4) θ – orientation of the selected agent;
5) θ_g – angle between the current position and goal of the selected agent;
6) ω – angular velocity of the selected agent;
7) g_{seq} – sequence of goal positions reached by the selected agent;
8) d_{obs} – distance between the selected agent's current position and the nearest obstacle.

In Step 2, we exploited the data extracted in Step 1 for the causal analysis. The main goal here was to reconstruct the causal model using our approach, F-PCMCI. For comparison, we repeated the causal analysis using the same data with PCMCI. Fig. 18 shows the two causal models derived from PCMCI and F-PCMCI relatively to agent 11 of the THÖR dataset. As expected, due to the large number of variables and links, the PCMCI algorithm is affected by spurious links, which it is not able to filter out. In contrast, F-PCMCI provides a simpler and more realistic causal model. It includes the full set of variables (like PCMCI) but retains only the most meaningful links between them, thanks to the TE-based filtering step. The execution time of the causal discovery confirmed our hypothesis: PCMCI completed in 79'45", while the F-PCMCI's execution lasted only 17'33", i.e. more than 4 times faster. A qualitative discussion about the correctness of the reconstructed causal models can be found in [26].

Lacking a ground truth model, we had to assess the correctness and accuracy of our causal model by evaluating the prediction accuracy of the causality-augmented architecture described in the following. In Step 3, we implemented an LSTM-based encoder-decoder model for Multi-Output Multi-Step forecasting[6]. The architecture, inspired by the DA-RNN network used in Section 2.3 and discussed in [14, 6, 28], was adapted to leverage the causal knowledge derived from our approach. Specifically, the self-attentions mechanism of the encoder was adapted to embed the causal inference vector from the discovered causal model as a non-trainable parameter. This allows the network to prioritise causally relevant drivers during prediction.

[6] https://github.com/lcastri/cmm_ts

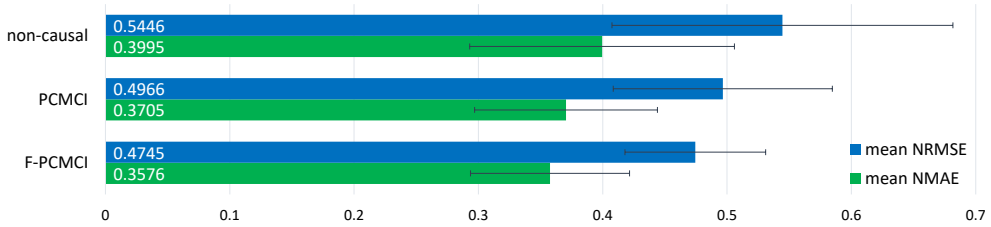


Figure 19: Comparison between non-causal and causal prediction (with both PCMCI and F-PCMCI) using mean NMAE and mean NRMSE across all the agents in the scenario. White numbers and error bars indicate mean and standard deviation, respectively.

The model was trained and tested using a 70%-10%-20% split of the time-series dataset. Hyperparameters, such as learning rate, batch size, and LSTM cell count, were optimized via grid search. Observation and forecasting windows were set to 32 (3.2s) and 48(4.8s) time steps, respectively. A separate network was trained for each agent in the scenario and tested on others to ensure robustness. To evaluate the quality of our causality-enhanced prediction, we used the *Normalised Mean Absolute Error* (NMAE) and the *Normalised Root Mean Square Error* (NRMSE), defined in [26]. Fig. 19 reports the comparison between prediction accuracy in the three cases: non-causal prediction, PCMCI-based prediction, and F-PCMCI-based prediction. The NMAE and NRMSE values are computed for each selected agent and then averaged. The figure clearly shows that the knowledge of the causal model helps to obtain a more accurate prediction. Moreover, since both errors are lower for the F-PCMCI's case compared to the PCMCI's one, we can conclude that our approach produces a better and more useful causal model.

In this activity, we extended and improved the state-of-the-art causal discovery algorithm PCMCI by embedding an additional feature-selection module based on transfer entropy. The proposed method was initially evaluated on two toy problems and on synthetic data from brain networks, where the ground-truth models were known a priori, to verify the correctness of the approach. It was then tested on a real-world robotics dataset containing large-scale time-series of human trajectories. We demonstrated that our approach significantly improves the PCMCI causal discovery method in terms of both accuracy and computational efficiency, enabling faster and more accurate causal discovery of dynamic models from real-world sensor data.

This activity resulted in the paper titled "Enhancing Causal Discovery from Robot Sensor Data in Dynamic Scenarios", accepted at the Conference on Causal Learning and Reasoning (CLeaR) [26]. Additionally, this work produced a Python implementation of the F-PCMCI algorithm[a] and the causality-augmented architecture of an LSTM-based encoder-decoder model for Multi-Output Multi-Step forecasting[b].

[a] https://github.com/lcastri/fpcmci
[b] https://github.com/lcastri/cmm_ts

3.3 Causal Discovery with Observational and Interventional Data from Time-Series

In Section 3.2 we introduced F-PCMCI to reduce the computational cost of causal discovery analysis. In this part of the deliverable, we describe the activities carried out to address a critical challenge for the causality and robotics communities: performing causal analysis that incorporates data from interventions. Observational data alone are often insufficient

to accurately identify the correct causal model in complex scenarios, where taking into account all the variables that influence the evolution of a system is not feasible. In such cases, *interventional data* – i.e., data obtained from controlled experiments – are necessary for causal discovery to eliminate spurious correlations and enhance the quality of the inferred causal model.

The proposed solution extends and improves the state-of-the-art causal discovery algorithm for time-series called Latent-PCMCI (LPCMCI) [29], taking inspiration from the way Joint Causal Inference (JCI) [30] handles interventions with known target. The result is a new algorithm that enables precise causal analysis using both observational and interventional data, which significantly improves the accuracy of the model discovered.

3.3.1 Interventions Through Context Variables

Combining observational and interventional data in causal discovery requires adapting the causal structure to account for both scenarios. Observational data considers the parents of the intervention variable, whereas interventional data necessitates breaking all incoming links to the intervention variable. To address this, we used context nodes inspired by the JCI framework [30]. Context nodes enable the integration of observational and interventional data into a unified causal structure while preserving system dependencies.

Meta-System Representation – Our approach models the system and context variables as a new meta-system $\mathscr{M}$, defined as:

$$\mathscr{M} : \begin{cases} X_i(t) = \tilde{f}(Pa(X_i), CX_k) & i \in \mathscr{I}, k \in \mathscr{K} \\ CX_k = f_k & k \in \mathscr{K} \end{cases} \tag{8}$$

where $\mathscr{I}$ represents the set of system variables defined as $X = (X_i)_{i \in \mathscr{I}}$, while $\mathscr{K}$ represents the set of context variables defined as $C = (CX_k)_{k \in \mathscr{K}}$. $Pa(X_i)$ is the parent set of the system variable X_i, instead CX_k is the context variable k. Moreover, the function $\tilde{f}$ models the system variables and can be decomposed as follows:

$$\tilde{f}(Pa(X_i), CX_k) : \begin{cases} f(Pa(X_i)) & CX_k = 0 \\ CX_k & CX_k = \xi_k \end{cases} \tag{9}$$

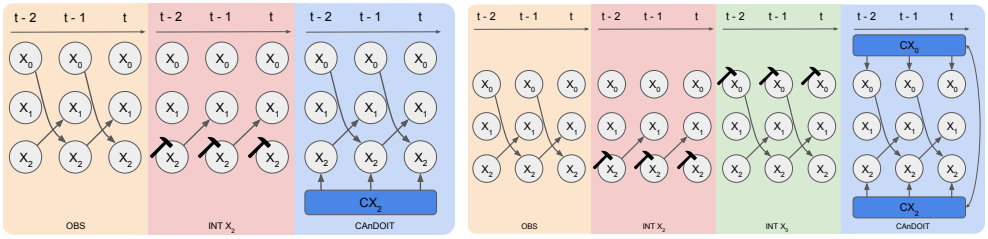
where f represents the function that models the evolution of the system variable X_i in the observational case, which depends solely on its parent set $Pa(X_i)$. Referring again to Equation 8, the function f_k models the context variables and it is defined as follows:

$$f_k : \begin{cases} \xi_k & \text{interventional mode for } k \\ 0 & \text{observational mode for } k \end{cases} \tag{10}$$

The case where $CX_k = 0$ corresponds to no intervention, i.e the observational baseline, while $CX_k = \xi_k$ models the intervention case, where ξ_k is the actual intervention value assigned to the variable X_i through the context variable CX_k.

Key Assumptions – In our case, to model hard interventions with known targets as context changes in time-series data, we adopt the "JCI123" framework [30], which relies on the following assumptions:

- **JCI1** *Exogeneity*: No system variable causes any context variable.
- **JCI2** *Complete randomised context*: No context variable is confounded with a system variable.



(a) **(b)**

Figure 20: CAnDOIT effectively employs context variables to handle observational and interventional data, resulting in a unified causal structure that accommodates both types of data. CAnDOIT blocks in a,b) provide examples of this unified causal structure in cases of single and multiple interventions, respectively.

JCI3 *Generic context model*: Context variables are interconnected but do not model causal relationships among themselves.

Additionally, we assume that each context node influences only one system variable, simplifying the integration of interventions with known targets.

At this point, we can further clarify the concept of context nodes. Essentially, the context node is a dummy exogenous variable (i.e. a "meta-variable" that does not exist in the real system) that is used to inject the interventional data into the intervention variable without ignoring its parents. Since the context node is exogenous (by the JCI1 assumption) and models the intervention, the model's structure does not change when transitioning between observational and interventional cases. Following the JCI framework for causal discovery with both observational and interventional data [30]. the interventional process in CAnDOIT is generated by creating context nodes (e.g. CX_k) that are added as parent of the system variables (e.g. X_k) and govern their possible values. The context node affects its system variable instantaneously (at the same time step) injecting the interventional data into it and maintaining its value constant for the duration of the intervention. Note that, as the context node does not carry temporal information, i.e., its value does not change over time, we modelled it as a unique node in the graph confounding its corresponding system variable at all the time intervals (see Fig. 20 CAnDOIT blocks).

Fig. 20a shows an example of a context variable to handle a hard intervention and illustrates how CAnDOIT creates an unified causal structure that represents both observational and interventional data.

3.3.2 CAnDOIT Algorithm

Fig. 21 depicts a detailed flowchart of CAnDOIT, explaining each step of the algorithm with an example. In particular, the steps executed by our approach are as follows:

- CAnDOIT takes observational and interventional data as input;
- Using the knowledge of the intervention target Z, the *context* block adds the context node CZ to the set of variables considered in $\mathscr{M}$, plus an instantaneous link $CZ \to Z$ to the initial causal structure, i.e., a fully connected graph that is the starting point of the LPCMCI algorithm;
- The system variables (X, Y, Z), along with the context node CZ are injected into the causal discovery block;
- LPCMCI performs the causal analysis on the meta-system $\mathscr{M}$ and then removes both the context variable CZ and the link $CZ \to Z$ before returning the causal model;

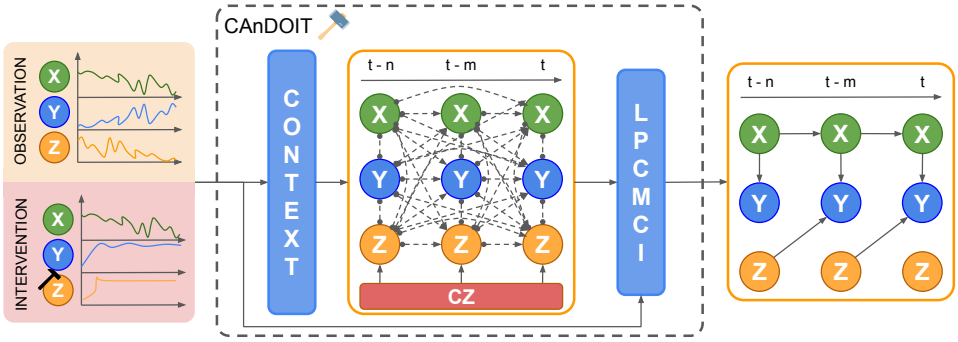


Figure 21: CAnDOIT's block scheme representation. CAnDOIT processes observational and interventional data; the context block adds context variables (CZ) linked to the actual intervention variable (Z) with an instantaneous link ($CZ \to Z$); Finally, the LPCMCI block finalises the causal discovery process.

- CAnDOIT outputs a time-series Partial Ancestral Graph (PAG).

Further details regarding the choice of LPCMCI and how CAnDOIT addresses the *faithfulness assumptions* are provided in the paper resulting from this activity [31]. A detailed pseudo-code explanation of our approach is presented in Algorithm 2. A Python implementation of CAnDOIT is also publicly available[7].

Algorithm 2 CAnDOIT

Require: obs. D_{obs} and int. D_{int} ts data, int. target variables X_i, significance level α, min and max time lag τ_{min}, τ_{max}.

$CM_0 \leftarrow$ fully connected PAG with $\circ\!\!\to$ for lagged dependencies and $\circ\!\!-\!\!\circ$ for contemporaneous dependencies ← LPCMCI starting point
$\mathscr{M} \leftarrow \mathscr{I}$ add the set of system variables $X = (X_i)_{i \in \mathscr{I}}$ to the meta-system $\mathscr{M}$
for each int. target variables X_i **do**
  $CX_k \leftarrow$ create the context variable CX_k associated to the intervention system variable X_i
  $\mathscr{M} \leftarrow CX_k$ add CX_k to the meta-system $\mathscr{M}$
  $CM_0 \leftarrow$ add CX_k to the LPCMCI initial condition CM_0
  for each τ in range(τ_{min}, τ_{max}) **do**
    $CM_0 \leftarrow$ add the link $CX_k \to X_i(t-\tau)$ to CM_0
$D_s \leftarrow [D_{obs}, D_{int}]$
$CM =$ `LPCMCI`$(D_s, \alpha, \tau_{min}, \tau_{max}, CM_0)$
$CM \leftarrow$ remove context variables CX_k and related links
return CM ← time-series PAG

Being based on LPCMCI, our CAnDOIT inherits its necessary conditions for proper functioning: Causal Markov Condition, Faithfulness, Acyclicity. Furthermore, like its predecessor, CAnDOIT can adapt to any type of data, including linear and nonlinear relationships, multiple time lags, various types of noise, and it cannot detect cyclical relationships. It retains the output format of a time-series Partial Ancestral Graph (PAG). Specifically, CAnDOIT produces a time-series PAG with a number of layers determined by the algorithm parameters τ_{min} and τ_{max} (see Algorithm 2 inputs). By default, τ_{min} is set to 0 to account for the instantaneous links created for the context variables. On the other hand, τ_{max} represents the maximum time delay considered when the algorithm performs conditional independence tests between variables across different time steps.

[7] https://github.com/lcastri/causalflow

Consequently, the time-series PAG consists of $\tau_{max}+1$ layers, corresponding to the time steps $t-\tau_{max}, t-(\tau_{max}-1), \ldots, t$.

PAGs are used to represent the Markov equivalence class of Maximal Ancestral Graphs (MAGs). The latter extend the DAGs representation by including also the bidirected link ($\leftrightarrow$) to represent variables confounded by a latent confounder. PAGs further generalise MAGs by incorporating additional edge types, specifically $\circ\!\!\rightarrow$ and $\circ\!\!-\!\!\circ$, to handle uncertainties in edge orientations. For example, in a PAG, a link $X \circ\!\!\rightarrow Y$ corresponds to two possible MAGs: $X \rightarrow Y$ (where X is an ancestor of Y) or $X \leftrightarrow Y$ (where X and Y are confounded by a latent variable). Similarly, a link $X \circ\!\!-\!\!\circ Y$ in a PAG represents two possible MAGs: $X \rightarrow Y$ (where X is an ancestor of Y) or $Y \rightarrow X$ (where Y is an ancestor of X).

3.3.3 Experiments

Our evaluation strategy was divided into two main parts. In the first part, we evaluated CAnDOIT's effectiveness in handling interventional data and its impact on the causal structure. Five testing strategies were devised, denoted as S_1, S_2, S_3, S_4, and S_5.

- In S_1, we assessed the performance of our approach with linear systems while varying the number of observable variables and without hidden confounders.
- S_2 extended S_1 by introducing hidden confounders while retaining linear systems and maintaining the same range of variables. In both S_1 and S_2, only a single intervention was conducted.
- In S_3, we evaluated CAnDOIT's performance with linear systems and hidden confounders when multiple interventions were applied, keeping the number of observable variables fixed.
- S_4 and S_5 mirrored S_2 and S_3, respectively, but focused on nonlinear systems.

Across all five evaluation strategies, CAnDOIT outperformed LPCMCI in terms of accuracy and uncertainty of the retrieved causal model. A full description and detailed results of these analyses are available in [31].

In the second part of our evaluation, we applied CAnDOIT to model a robotic scenario in a simulated environment. The strategy involved extracting time-series data from the simulator and performing causal discovery in the presence of a hidden confounder.

Causal World for Robot Camera Modelling – We designed an experiment to learn the causal model in a hypothetical robot arm application equipped with a camera. Our focus was on estimating the causal relationship between the color's brightness of objects as captured by the camera and various factors, including camera-to-object distance. For this evaluation, we utilised the well-known benchmark *Causal World* [32], which is designed for causal structure learning in a robotic manipulation environment. The environment consists of a TriFinger robot (shown in Fig. 22a and 22b), a floor, and a stage. It allows for the inclusion of objects with various shapes, e.g. cubes.

For simplicity, we focused on a specific scenario using only one finger of the robot, where the finger's end-effector was equipped with a camera. The scenario (shown in Fig. 22) consists of a cube placed at the centre of the floor, surrounded by a white stage. The color's brightness (b) of the cube and the floor is modelled as follows:

$$b = K_h \frac{H}{H_{max}} + K_v \left(1 - \frac{v}{v_{max}}\right) + K_d \frac{d_c}{d_{c_{max}}} \tag{11}$$

where H is the end-effector height, v its absolute velocity, and d_c the distance between the end-effector and the cube. K_h, K_v, K_d are the gains associated to each factors, while H_{max}, v_{max}, and $d_{c_{max}}$ are the maximum values for H, v, and d_c, respectively. This model

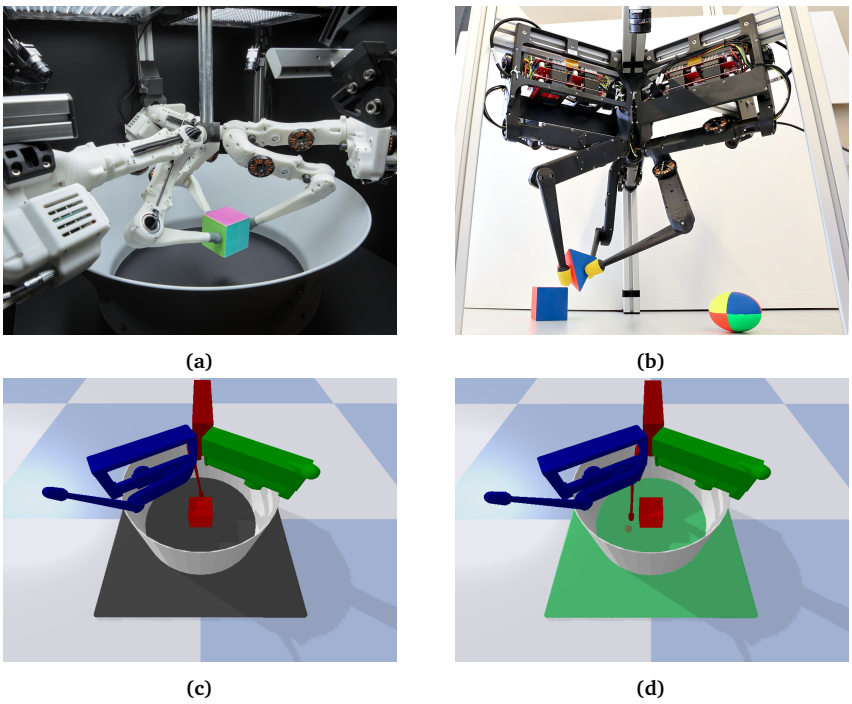


Figure 22: (a, b) TriFinger robot manipulating different objects. (c, d) CausalWorld [32]: a robotic manipulation simulator featuring the TriFinger robot. (c) Observational experiment; (d) Experiment with an intervention on the floor's color.

captures the shading and blurring effects on the cube due to the height of the end-effector, its velocity, and its distance from the cube. On the other hand, the floor, being darker and larger than the cube, is only affected by the end-effector's height.

The data collected from the scenario therefore includes the floor (F_c) and the cube (C_c) colors, as well as the height (H), the absolute velocity (v) of the end-effector, and its distance from the cube (d_c). The ground-truth structural causal model for the variables F_c and C_c is expressed as follows:

$$\begin{cases} F_c(t) = b(H(t-1)) \\ C_c(t) = b(H(t-1), v(t-1), d_c(t-1)) \end{cases} \tag{12}$$

Note that H, v, and d_c are obtained directly from the simulator and not explicitly modelled.

Experimental Results on Robotic Scenario – The evaluation involved three main steps. *(i)* We generated observational data containing all the variables in the system (F_c, C_c, H, v, d_c), as shown in Fig. 22c, and performed the causal analysis using LPCMCI. *(ii)* We intentionally hid the variable H, representing the height of the end-effector, to create a hidden confounder and a spurious relationship between C_c and F_c. Again, we used LPCMCI for the causal analysis. *(iii)* We conducted an intervention on the floor's color, setting it to green (Fig. 22d), and collected data from the simulator. Then we used CAnDOIT for the causal analysis with both observational and interventional, accounting for the hidden confounder H. The observational time-series had a length of 600 samples, while the interventional time-series consisted of 125 samples. Both were recorded at a sampling rate of 10Hz. Also in this case, to ensure a fair analysis, LPCMCI and CAnDOIT used exactly the same amount of data. Consequently, LPCMCI received the complete set

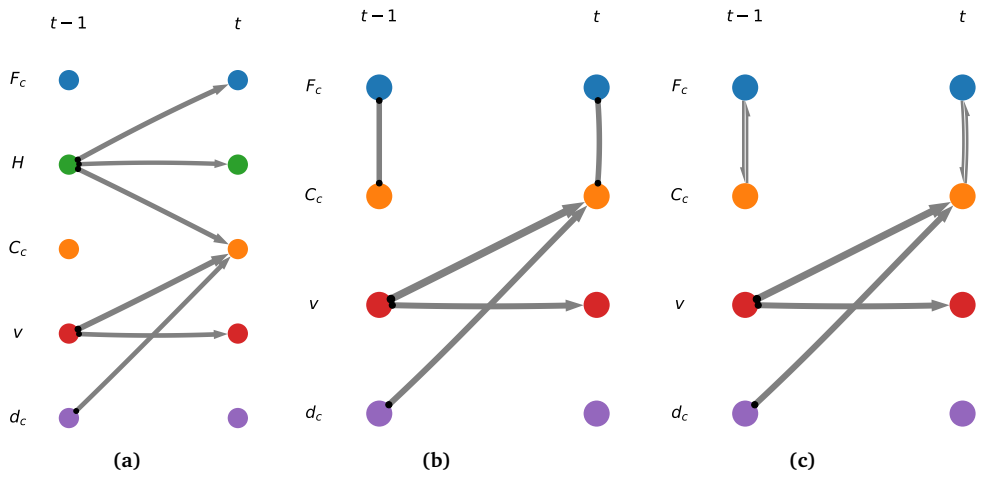


Figure 23: Causal model of the robot camera in Causal World: (a) LPCMCI's result with all the variables being observable; (b) LPCMCI's result with hidden H; (c) CAnDOIT's causal model with hidden H.

of observational data, whereas for CAnDOIT part of the observational data was replaced by interventions, specifically 475 observational samples and 125 interventional ones.

Fig. 23 shows the results for each specific step: (Fig. 23a) causal model using LPCMCI with observable variables only; (Fig. 23b) LPCMCI's result with hidden H; (Fig. 23c) causal model retrieved by CAnDOIT, leveraging both observational and interventional data (generated by the simulations shown in Fig. 22c and 22d) and successfully identifying the bidirected relation between C_c and F_c, which represents the presence of a latent confounder (H).

Also in this experiment, we can see the benefit of using intervention data alongside the observations. In Fig. 23b, LPCMCI is not able to orient the contemporaneous (spurious) link between F_c and C_c due to the hidden confounder H. This yields the ambiguous link $F_c \circ\!\!-\!\!\circ C_c$, which does not encode the correct link $\longleftrightarrow$ (the $\circ\!\!-\!\!\circ$ represents either $\rightarrow$ or $\leftarrow$). Instead CAnDOIT, using interventional data, correctly identifies the bidirected link $F_c \longleftrightarrow C_c$, decreasing once again the uncertainty level and increasing the accuracy of the reconstructed causal model.

In this activity we implemented CAnDOIT, a new state-of-the-art algorithm that enables causal discovery using both observational and interventional data via context variables. This advancement addresses a critical need in both the robotics and causality communities, where understanding cause-and-effect relationships is essential for building intelligent, adaptive systems. We validated our approach experimentally on random synthetic models and tested on a robotic simulator for causal discovery, focusing on the significance of interventional data. Our results confirmed that CAnDOIT outperforms previous causal discovery methods, improving accuracy and enhancing model identifiability. They also highlight its capability to handle interventional data effectively, and its potential benefit for real-world robot applications. The proposed method lays the foundation for new observations- and interventions-based causal discovery methods on time-series data, with numerous opportunities for future research.

This activity resulted in the journal paper titled "CAnDOIT: Causal Discovery with Observational and Interventional Data from Time Series", published on Advanced Intelligent Systems [26]. Additionally, this work produced a publicly-available Python implementation of the CAnDOIT algorithm[a].

[a] https://github.com/lcastri/causalflow

3.4 A ROS-based Causal Framework for Human-Robot Interaction Applications

In the activity presented in Section 3.2, we extended the state-of-the-art causal discovery algorithm PCMCI by embedding an additional feature-selection module based on transfer entropy. This step was necessary to reduce the computation cost of the causal analysis due to the number of variables, which is a significant challenge for autonomous robots that often operate with limited hardware resources and real-time constraints.

However, many causal discovery methods, including F-PCMCI, lack the capability to run directly on the robot. This limitation pose challenges for exploiting the reconstructed causal models in real-time. In particular, due to the aforementioned limitation, a robot must accumulate a significant amount of data and then conduct offline causal analysis. Subsequently, the reconstructed causal model has to be reintegrated into the robot for utilisation. The limitation may stem from the lack of a software framework that facilitates the integration between the two communities (i.e. robotics and causality) and that operates directly inside the Robot Operating System (ROS)[8], the standard de facto in robotics. The solution presented in this part of the deliverable aims to streamline this process by enabling the robot to conduct onboard causal discovery on data batches while concurrently collecting data for future causal analysis. Moreover, given the integration of our framework within ROS, the acquired causal model can be directly exploited by the robot for reasoning and planning problems.

3.4.1 ROS-based Causal Analysis Framework

The proposed approach, named ROS-Causal, extracts and collects data from an HRI scenario, such as agents' trajectories, and then performs causal analysis on the collected data in a batched manner. A modular ROS Python library implementation of ROS-Causal has been developed and made publicly available[9]. The modular design allows for the expansion of the library with new causal discovery methods. In the following, we provide

[8] https://www.ros.org/
[9] https://github.com/lcastri/roscausal.git

a detailed explanation of the three main blocks that compose the ROS-Causal pipeline, as depicted in Fig. 24. Information regarding subscribers and publishers for each ROS node is summarised in Table 7.

Data Merging – The purpose of this block is to merge robot and human data from various topics into custom ROS messages in the ROS-Causal framework. The nodes `roscausal_robot` and `roscausal_human` extract the position, orientation, velocities and target positions of the robot and the human, respectively. These data are retrieved from ROS topics/params relative to the robotic platform and need to be configured within the framework. Then, the two nodes merge the acquired data into the ROS messages `RobotState` and `HumanState` published on the predefined topics `/roscausal/robot` and `/roscausal/human`. The latter are utilised in the data collection block explained in the following section.

Data Collection and Post-processing – This block takes input from the previous block's topics to create a data batch for the causal discovery node. More in detail, the `roscausal_data` node subscribes to the topics `/roscausal/robot` and `/roscausal/human` and begins collecting data in a CSV file. Once the desired time-series length, configurable as a ROS parameter, is reached, the node provides the option to post-process the data, allowing for the creation of a high-level representation of the scenario. For instance, from the low-level data, such as agents' trajectories, a post-processing script can be specified to generate distances and angles between the agents. Once the post-processing is complete, the CSV file is saved into a designated folder (e.g. "csv_pool" in Fig. 24).

Causal Discovery – The ROS node `roscausal_discovery` performs

Table 7: ROS-Causal subscribers and publisher.

`roscausal_robot`		
subscribed topics	**description**	**msg type**
to be setup	robot pose	to be setup
to be setup	robot velocity	to be setup
to be setup	robot goal	to be setup
published topics	**description**	**msg type**
`/roscausal/robot`	full robot state	RobotState
`roscausal_human`		
subscribed topics	**description**	**msg type**
to be setup	human pose	to be setup
to be setup	human velocity	to be setup
to be setup	human goal	to be setup
published topics	**description**	**msg type**
`/roscausal/human`	full human state	HumanState
`roscausal_data`		
subscribed topics	**description**	**msg type**
`/roscausal/robot`	full robot state	RobotState
`/roscausal/human`	full human state	HumanState
`roscausal_discovery`		
published topics	**description**	**msg type**
`/roscausal/causal_model`	causal model description	CausalModel
`/roscausal/dag`	Summary DAG	Image
`/roscausal/tsdag`	Time-series DAG	Image

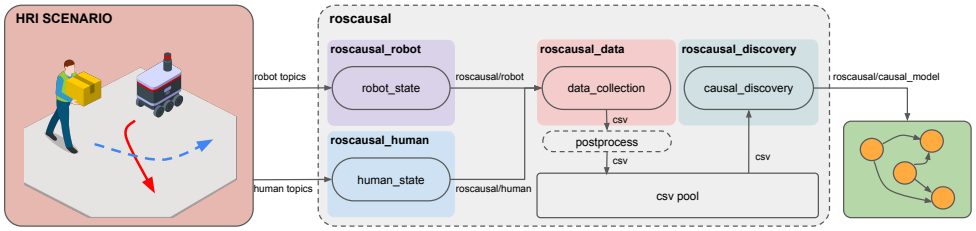


Figure 24: ROS-Causal pipeline: *(i)* data extraction from human-robot interaction scenarios; *(ii)* collection and post-processing of data to derive a high-level representation of the scenario. *(iii)* causal discovery conducted on the extracted data, with the resulting causal model published on a dedicated rostopic.

causal discovery analysis on the collected data. Specifically, it continuously checks for the presence of a CSV file in the designated folder. Upon locating a file, it initiates the causal analysis on that specific data batch. It is important to note that the `roscausal_data` and `roscausal_discovery` ROS nodes operate asynchronously, allowing the simultaneous execution of causal analysis on one dataset while continuing the collection of another. The `roscausal_discovery` ROS node incorporates two causal discovery methods: the PCMCI [18] and its extension, F-PCMCI presented in Section 3.2. For both algorithms, the following parameters, handled as ROS parameters, needs to be set: *(i)* significance threshold (typically $\alpha = 0.05$); *(ii)* minimum and maximum time lag; *(iii)* conditional independence test. Once the causal analysis is complete, `roscausal_discovery` ROS node deletes the just examined CSV dataset in order to maintain robot's memory free and decomposes the causal model into three `n.lags` × `n.vars` × `n.vars` matrices. Here, `n.lags` represents the number of time lags to the current time where causal dependencies are tested, defined as the difference between the maximum and minimum time lag, and `n.vars` represents the number of variables. Each matrix contains distinct information about the built causal model for each time lag (further details can be found in [33]). The three matrices are embedded in the `CausalModel` ROS message and published on `/roscausal/causal_model`, enabling access across the robotic system. Visualisations of the causal model, as a summary DAG and time-series DAG, are published as `Image` messages on `/roscausal/dag` and `/roscausal/tsdag`.

3.4.2 Human-Robot Interaction Simulator

To assess the effectiveness of our approach in reconstructing causal models from HRI scenarios, we developed a dedicated Gazebo-based simulator called ROS-Causal_HRISim. This simulator accurately mimics HRI scenarios involving a TIAGo[10] robot and multiple pedestrians modelled using the *pedsim_ros*[11] ROS library. The latter simulates individual and group social activities (e.g., walking) using a social force model. To better emulate human behaviours, we incorporated the option for user teleoperation (via keyboard) of a simulated person, not influenced by social forces. A Docker image of ROS-Causal_HRISim, comprising also ROS-Causal, has been created and is publicly available[12]. An HRI scenario created by ROS-Causal_HRISim is shown in Fig. 25.

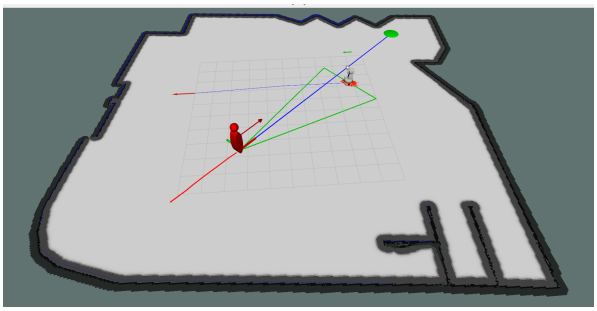


Figure 25: HRI scenario involving a TIAGo robot and a teleoperated person, created by ROS-Causal_HRISim.

3.4.3 Experiments

Our evaluation strategy consisted of two steps. First, we validated the correctness and effectiveness of ROS-Causal in a simulated HRSI environment. This step was crucial for assessing ROS-Causal's capability to reconstruct the correct causal model from data before deploying it on the real robot. Second, we evaluated ROS-Causal in a real HRSI scenario,

[10] https://pal-robotics.com/robots/tiago/
[11] https://github.com/srl-freiburg/pedsim_ros
[12] https://github.com/lcastri/ROS-Causal_HRISim

where data collection and causal discovery were performed directly on the real robot. The experiments have been designed to investigate the following research questions:

- R_1) Is it feasible to generate causal models onboard the robot via ROS-Causal?
- R_2) If yes, how much data (i.e., time-series length and sampling frequency) are needed to generate accurate causal models?
- R_3) If yes, how much execution time the generation takes?

The HRSI scenario chosen for the evaluation strategy takes inspiration from the human-moving obstacles scenario, which we analysed in Section 3.1.1. The scenario depicts a typical DARKO setting, where a person and a robot deliver parcels at different target stations. The person has to reach a predefined target position, which dynamically changes when reached, and avoid the robot that crosses his/her path. The robot follows a predetermined path along its targets. When the person encounters the robot, he/she must avoid it by decreasing his/her velocity and/or adjusting his/her steering. In addition, as the person approaches the target position, he/she gradually reduces the velocity.

The set of variables used to model this scenario and the set of causal links between them aligns with what already explained in Section 3.1.1. The robot was perceived as an obstacle by the person. However, ROS-Causal can be further applied to various scenarios involving robots and humans, such as a robot following a person or interactive tasks between them.

ROS-Causal Simulation Evaluation – Our plan was to create the HRSI scenario just discussed, collect the trajectories of the two agents (i.e., robot and person), process the collected data to obtain the desired set of variables previously discussed (v, d_g, r) and finally execute the causal discovery on it. Fig. 25 shows the HRSI scenario created by ROS-Causal_HRISim. It involves a TIAGo robot and a simulated person teleoperated by one of the participants via keyboard, represented by the red manikin. The green dot represents the person's target position, while the blue line visualises the distance between the person and his/her goal position. Finally, the green cone is the visualisation of the collision risk. It is built from the person position to the enlarged encumbrance of the TIAGo, which is perceived by the human as a moving obstacle.

Regarding the ROS-Causal parameters and settings used for the data collection and causal analysis, we configured a desired time-series length corresponding to a timeframe of 150s and recorded the trajectories of the two agents, their linear velocity, and orientation, with a sampling frequency of 10Hz. Subsequently, we compute the distance between the human and the goal, as well as the risk of collision. For the causal discovery block, we employed the F-PCMCI causal discovery method with a significance level of $\alpha = 0.05$, a conditional independence test based on Gaussian Process regression and Distance Correlation (GPDC). We also used a 1-step lag time, meaning variables at time t could only be affected by those at time $t-1$. The resulting causal model is depicted in Fig. 27a. The graph faithfully represents the expected model discussed earlier and is consistent with the result in Fig. 14 (right).

Dataset and Experimental Setup – After confirming the correct functionalities of the ROS-Causal framework, we proceeded with the lab evaluation analysis, replicating the scenario staged through ROS-Causal_HRISim in the lab environment, as shown in Fig. 26a. The experiment and data collection occurred in a laboratory room of $5 \times 8.2m$ in Fig. 26b. Fifteen participants (6 females), aged between 25 and 55, took part in the experiment. Seven of them were researchers who regularly work with robots. Only point cloud readings from the Velodyne VLP-16 3D LiDAR were recorded.

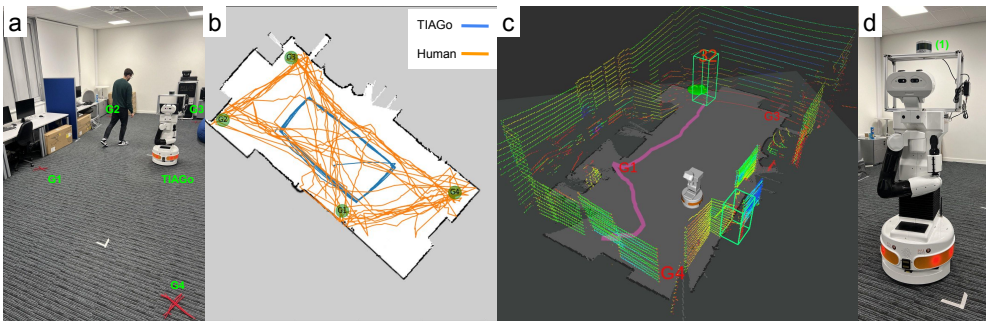


Figure 26: (a) HRSI experiment in a lab scenario with a TIAGo robot, a person and his/her four goal positions;(b) 2D map of an experiment with a person and TIAGo, with trajectories in orange and blue respectively, and four goal positions (green dot); (c) RViz visualisation of the scenario; (d) TIAGo robot with (1) a Velodyne VLP-16 3D LiDAR used for dataset collection.

Participants' task was to navigate between four designated goal positions while avoiding collisions with the robot when crossing paths. Specifically, they were instructed to begin from a goal position randomly chosen by themselves, select and walk towards the next one, also randomly chosen, and repeat this process until the robot stopped (i.e., after 5 minutes from the start). They were asked to pass through all the goal positions at least 7 times, avoiding the robot when they encountered it. No specific instructions were provided on how to reach the goals or avoid the robot.

A predefined rectangular path (i.e., in blue in Fig. 26b) was set for the TIAGo robot to navigate along the room and generate frequent interactions with the participants. As mentioned earlier, in this experimental setting, the robot was considered by the participant as an obstacle to avoid while walking towards their target positions. Fig. 26a shows an example of the experiment, while Fig. 26b shows the trajectories of the two agents (i.e., the ones related to the robot in blue and the human in orange).

To track the motion of the agents, we used a Velodyne VLP-16 3D LiDAR and the Bayes People Tracker[13] [15] on the related point cloud. Fig. 26c illustrates, through RViz, the human tracked by the robot. The robot equipped with the Velodyne is shown in Fig. 26d. More details about the data collection process and the dataset resulted from this activity can be found in [34].

ROS-Causal Evaluation in Lab Scenario – Data collection, post-processing, and causal discovery were all executed by our ROS-Causal framework with the same parameters used for the simulation, as explained in the ROS-Causal Simulation Evaluation paragraph. Fig. 27b shows the causal model relative to one of the participants. The graph accurately represents the expected model discussed in the Human-moving obstacles scenario paragraph of Section 3.1.1 and is consistent with the result presented in Fig. 14 (right), as well as with the model obtained from the simulation experiment in Fig. 27a. This demonstrates the reliability of ROS-Causal_HRISim to mimic HRI scenarios and the ROS-Causal's ability to retrieve the expected causal model in both simulation and lab experiments, validating the onboard causal discovery via ROS-Causal (R_1).

Sampling Frequency analysis – Fig. 27c shows the impact of sampling frequency on causal discovery. We analysed time-series data at varying frequencies, from 0.5Hz to the original 10Hz. The Structural Hamming Distance (SHD) – a standard distance to compare graphs by their adjacency matrix – of the retrieved causal models, compared to

[13] https://github.com/LCAS/bayestracking

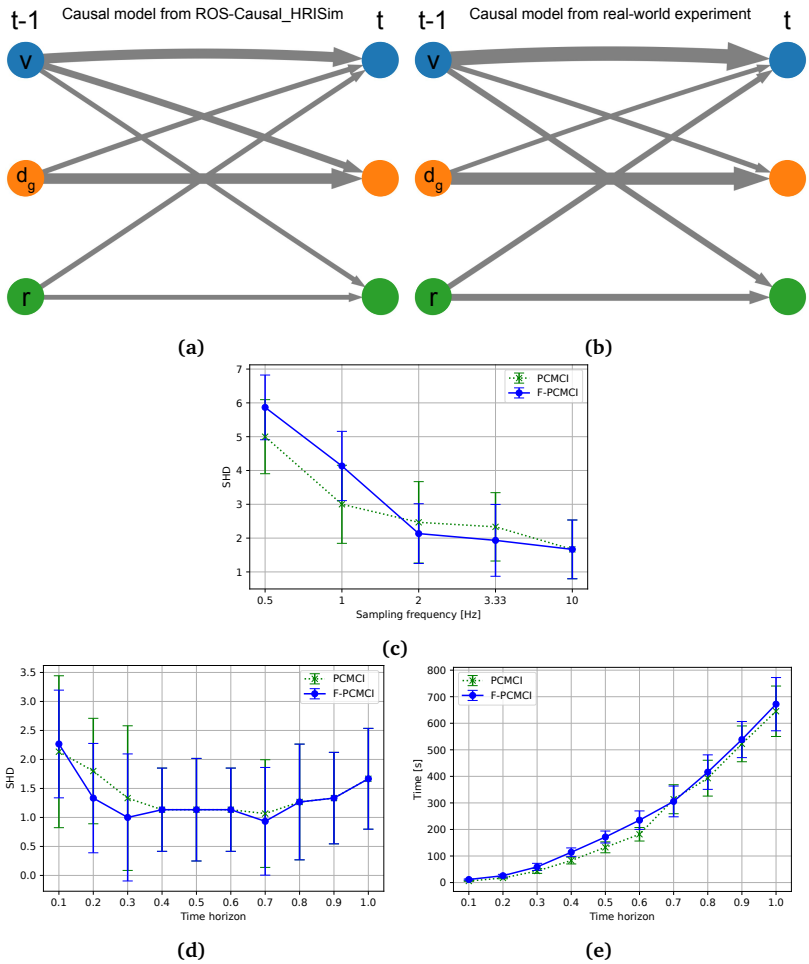


Figure 27: (a), (b) Causal model reconstructed by ROS-Causal from simulation and lab experiments, respectively. Execution time (c) and SHD (d) analyses with various time horizons. (e) SHD analysis based on the sampling frequency.

the baseline, was calculated for each frequency. The results indicate that the original 10Hz frequency is crucial for obtaining accurate causal models.

Time-Horizon analysis – Fig. 27d and 27e present the SHD and execution time for different time horizons. We tested time-series lengths ranging from 10% to 100% of the full length, corresponding to approximately 5 minutes per participant. SHD values were measured against a baseline causal graph obtained through simulation (Fig. 27a). The results suggest that a time window between 30% and 70% of the full time-series length (i.e., 90 to 210 seconds) provides sufficient data for accurate causal model retrieval. Shorter time-series were insufficient for accurate model learning, while longer sequences risked overfitting due to the parametric kernel estimator used in the causal discovery. Our key findings can be summarised as follows:

- **Sampling Frequency** – The original 10Hz sampling rate is essential for generating accurate causal models, as shown in Fig. 27c.
- **Time-Series Length** – A time-series length between 30% and 70% of the full duration (roughly 90 to 210 seconds) is optimal for causal model accuracy, as illustrated in

Fig. 27d and 27e.

- **Optimal Trade-off** – A 40% length of the time-series (approximately 120 seconds) recorded at 10Hz strikes the best balance between model accuracy and execution time ($\sim$ 100s), addressing the research questions (R_2) and (R_3).

Further details about the results can be found in our paper related to this activity [34].

In this work, we introduced ROS-Causal, a ROS-based framework for causal analysis in human-robot spatial interaction (HRSI), and its complementary ROS-Causal_HRISim simulator. ROS-Causal facilitates onboard data collection and causal discovery, enabling robots to simultaneously reconstruct causal models while gathering data for future analysis. To evaluate the effectiveness of ROS-Causal in modelling these interactions, we designed identical HRSI scenarios in both ROS-Causal_HRISim and real-world lab environments. Causal discovery conducted on both setups produced consistent causal models, demonstrating the simulator's ability to replicate realistic HRSI scenarios. Our results validate the feasibility of onboard causal discovery using a real robot and provide insights into the simulator's capability to represent useful HRSI situations. Furthermore, we analysed the execution time and data requirements—specifically time-series length and sampling frequency—needed for generating accurate causal models in a given scenario.

This research work led to two publications:

- "ROS-Causal: A ROS-Based Causal Analysis Framework for Human-Robot Interaction Applications", presented at the Workshop on Causal Learning for Human-Robot Interaction (Causal-HRI), part of the ACM/IEEE International Conference on Human-Robot Interaction (HRI) [33].
- "Experimental Evaluation of ROS-Causal in Real-World Human-Robot Spatial Interaction Scenarios", presented at the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) [34].

This activity also produced several technical and software contributions:

- ROS-Causal[a]: A ROS-based causal analysis framework for HRI applications.
- ROS-Causal_HRISim[b]: An ad-hoc simulator for HRI to facilitate scenario design and collect both observational and interventional data for causal analysis.
- Causal HRSI Dataset[c]: A dataset for evaluating human-robot spatial interactions and enabling causal analysis using mobile platforms.

Finally, ROS-Causal is actively being used on the DARKO platform. Its capabilities and functionalities were successfully demonstrated during a live stakeholder meeting, where it reconstructed a causal model of a human-human spatial interaction scenario using perception and localisation data provided by WP2 and WP3.

[a] https://github.com/lcastri/roscausal
[b] https://github.com/lcastri/ROS-Causal_HRISim
[c] https://zenodo.org/records/10844902

4 Conclusions

This deliverable highlights the activities and advancements related to T5.3 and T5.4 in enhancing human-robot spatial interactions. In particular, in T5.3 we leveraged neuro-symbolic (QTC-based) approaches for enhancing context-aware human motion representation, while in T5.4 we developed new causal discovery methods for robotics applications to enable high-level reasoning on real sensor data. The activities detailed in this report contribute to advancing the fields of mobile robotics, human motion representation, and causal inference.

The document also presents the tools developed during these activities, namely *neuROSym*, *F-PCMCI*, *CAnDOIT*, and *ROS-Causal*, which are all publicly available and currently integrated into the DARKO platform.

References

[1] Matthias Delafontaine, Seyed Hossein Chavoshi, Anthony G Cohn, and Nico Van de Weghe. “A qualitative trajectory calculus to reason about moving point objects”. In: *Qualitative spatio-temporal representation and reasoning: Trends and future directions*. IGI Global, 2012, pp. 147–167.

[2] Nicola Bellotto, Marc Hanheide, and Nico Van de Weghe. “Qualitative design and implementation of human-robot spatial interactions”. In: *Social Robotics: 5th International Conference, ICSR 2013, Bristol, UK, October 27-29, 2013, Proceedings 5*. Springer. 2013, pp. 331–340.

[3] Nico Van de Weghe. “Representing and reasoning about moving objects: A qualitative approach”. PhD thesis. Ghent University, 2004.

[4] Marc Hanheide, Annika Peters, and Nicola Bellotto. “Analysis of human-robot spatial behaviour applying a qualitative trajectory calculus”. In: *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE. 2012, pp. 689–694.

[5] Christian Dondrup, Nicola Bellotto, and Marc Hanheide. “A probabilistic model of human-robot spatial interaction using a qualitative trajectory calculus”. In: *2014 AAAI Spring Symposium Series*. 2014.

[6] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W Cottrell. “A dual-stage attention-based recurrent neural network for time series prediction”. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2017, pp. 2627–2633.

[7] Edward Twitchell Hall and T Edward. “Hall. The hidden dimension”. In: *Anchor Books New York* 20 (1969), p. 71.

[8] Sariah Mghames, Luca Castri, Marc Hanheide, and Nicola Bellotto. “Qualitative Prediction of Multi-Agent Spatial Interactions”. In: *32nd IEEE International Conference on Robot and Human Interactive Communication*. Busan, South Korea, Aug. 2023.

[9] Nico Van de Weghe and Philippe De Maeyer. “Conceptual neighbourhood diagrams for representing moving objects”. In: *Perspectives in Conceptual Modeling: ER 2005 Workshops AOIS, BP-UML, CoMoGIS, eCOMO, and QoIS, Klagenfurt, Austria, October 24-28, 2005. Proceedings 24*. Springer. 2005, pp. 228–238.

[10] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. "Social gan: Socially acceptable trajectories with generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2255–2264.

[11] Andreas Ess, Bastian Leibe, and Luc Van Gool. "Depth and appearance for mobile scene analysis". In: *2007 IEEE 11th international conference on computer vision*. IEEE. 2007, pp. 1–8.

[12] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. "Crowds by example". In: *Computer graphics forum*. Vol. 26. 3. Wiley Online Library. 2007, pp. 655–664.

[13] Roberto Martín-Martín, Hamid Rezatofighi, Abhijeet Shenoi, Mihir Patel, J Gwak, Nathan Dass, Alan Federman, Patrick Goebel, and Silvio Savarese. "Jrdb: A dataset and benchmark for visual perception for navigation in human environments". In: *arXiv preprint arXiv:1910.11792* (2019).

[14] Sariah Mghames, Luca Castri, Marc Hanheide, and Nicola Bellotto. "A Neuro-Symbolic Approach for Enhanced Human Motion Prediction". In: *International Joint Conference on Neural Networks (IJCNN)*. Queensland, Australia, June 2023.

[15] Zhi Yan, Tom Duckett, and Nicola Bellotto. "Online learning for human classification in 3D LiDAR-based tracking". In: *IEEE/RSJ Int. Conf. on Intell. Robots & Systems (IROS)*. IEEE. 2017, pp. 864–871.

[16] Andrey Rudenko, Tomasz P Kucner, Chittaranjan S Swaminathan, Ravi T Chadalavada, Kai O Arras, and Achim J Lilienthal. "THÖR: Human-Robot Navigation Data Collection and Accurate Motion Trajectories Dataset". In: *IEEE Robotics & Automation Letters* (2020), pp. 676–682.

[17] Sariah Mghames, Luca Castri, Marc Hanheide, and Nicola Bellotto. "neuROSym: Deployment and Evaluation of a ROS-based Neuro-Symbolic Model for Human Motion Prediction". In: *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE. 2024.

[18] J. Runge. "Causal network reconstruction from time series: From theoretical assumptions to practical estimation". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* (2018), p. 075310.

[19] Qinghua Li, Zhao Zhang, Yue You, Yaqi Mu, and Chao Feng. "Data driven models for human motion prediction in human-robot collaboration". In: *IEEE Access* (2020), pp. 227690–227702.

[20] Paolo Fiorini and Zvi Shiller. "Motion planning in dynamic environments using velocity obstacles". In: *Int. Journal of Robotics Research* (7 1998).

[21] Dražen Brščić, Takayuki Kanda, Tetsushi Ikeda, and Takahiro Miyashita. "Person tracking in large public spaces using 3-D range sensors". In: *IEEE Trans. on Human-Machine Systems* (2013), pp. 522–534.

[22] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. "Detecting and quantifying causal associations in large nonlinear time series datasets". In: *Science Advances* (2019).

[23] Luca Castri, Sariah Mghames, Marc Hanheide, and Nicola Bellotto. "Causal discovery of dynamic models for predicting human spatial interactions". In: *International Conference on Social Robotics*. Springer. 2022, pp. 154–164.

[24] J. Runge. "Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 1388–1397.

[25] M. Wienöbst, M. Bannach, and M. Liśkiewicz. "Extendability of causal graphical models: Algorithms and computational complexity". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 1248–1257.

[26] Luca Castri, Sariah Mghames, Marc Hanheide, and Nicola Bellotto. "Enhancing Causal Discovery from Robot Sensor Data in Dynamic Scenarios". In: *2nd Conference on Causal Learning and Reasoning*. 2023.

[27] S. M Smith, K. L Miller, G. Salimi-Khorshidi, M. Webster, C. F Beckmann, T. E Nichols, J. D Ramsey, and M. W Woolrich. "Network modelling methods for FMRI". In: *Neuroimage* 54.2 (2011), pp. 875–891.

[28] X. Yin, Y. Han, H. Sun, Z. Xu, H. Yu, and X. Duan. "Multi-attention generative adversarial network for multivariate time series prediction". In: *IEEE Access* 9 (2021), pp. 57351–57363.

[29] Andreas Gerhardus and Jakob Runge. "High-recall causal discovery for autocorrelated time series with latent confounders". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12615–12625.

[30] Joris M Mooij, Sara Magliacane, and Tom Claassen. "Joint causal inference from multiple contexts". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 3919–4026.

[31] Luca Castri, Sariah Mghames, Marc Hanheide, and Nicola Bellotto. "CAnDOIT: Causal Discovery with Observational and Interventional Data from Time Series". In: *Advanced Intelligent Systems* n/a.n/a (), p. 2400181. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202400181.

[32] Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf, and Stefan Bauer. "CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning". In: *International Conference on Learning Representations*. 2021.

[33] Luca Castri, Gloria Beraldo, Sariah Mghames, Marc Hanheide, and Nicola Bellotto. "ROS-Causal: A ROS-based Causal Analysis Framework for Human-Robot Interaction Applications". In: *Workshop on Causal Learning for Human-Robot Interaction (Causal-HRI), ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2024.

[34] Luca Castri, Gloria Beraldo, Sariah Mghames, Marc Hanheide, and Nicola Bellotto. "Experimental Evaluation of ROS-Causal in Real-World Human-Robot Spatial Interaction Scenarios". In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. 2024, pp. 1603–1609.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017274