






# Cooperative Lidar Sensing for Pedestrian Detection: Data Association Based on Message Passing Neural Networks

Bernardo Camajori Tedeschini , *Graduate Student Member, IEEE*, Mattia Brambilla , *Member, IEEE*,  
Luca Barbieri , *Member, IEEE*, Gabriele Balducci ,  
and Monica Nicoli , *Senior Member, IEEE*

**Abstract**—This paper considers the problem of cooperative lidar sensing in vehicular networks. We focus on the task of associating the vehicle-generated measurements by lidars to enable a cooperative detection of vulnerable road users. The considered measurements are the three-dimensional bounding boxes extracted from the lidar point cloud. Focusing on a centralized architecture which aggregates and processes all the sensing information, we design a graph formulation of the association problem and we propose a novel solution based on Message Passing Neural Networks (MPNNs). The method has the advantage of accurately learning the associations and the measurement statistics directly from data. We validate the proposed approach on a cooperative sensing scenario simulated by CARLA, an open-source high-fidelity simulator for automated driving scenarios. For the generation of bounding boxes related to pedestrian detections, we consider both artificially-generated and realistic measurements obtained by employing the PointPillars model. We validate the performance by comparing the proposed MPNN model with the Sum-Product Algorithm for Data Association (SPADA), a common approach for data association in multisensor systems. The proposed data-driven MPNN model achieves an association accuracy above 99% and outperforms SPADA in case of moderate sensing errors, as foreseen by automated driving scenarios. We also assess the efficacy of data association in case of mis-modeling between training and testing datasets, observing good generalization capabilities when dealing with untrained conditions.

**Index Terms**—Cooperative lidar, pedestrian detection, data association, MPNN, SPADA, CARLA simulator.

Manuscript received 8 November 2022; revised 17 April 2023 and 31 July 2023; accepted 7 August 2023. Date of publication 14 August 2023; date of current version 5 September 2023. This work was supported by the project Centro Nazionale per la Mobilità Sostenibile (MOST), funded by the Italian Ministry of University and Research under the PNRR funding program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mojtaba Soltanalian. (*Corresponding author: Bernardo Camajori Tedeschini.*)

Bernardo Camajori Tedeschini, Mattia Brambilla, Luca Barbieri, and Gabriele Balducci are with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy (e-mail: [bernardo.camajori@polimi.it](mailto:bernardo.camajori@polimi.it); [mattia.brambilla@polimi.it](mailto:mattia.brambilla@polimi.it); [luca.l.barbieri@polimi.it](mailto:luca.l.barbieri@polimi.it); [gabriele.balducci@mail.polimi.it](mailto:gabriele.balducci@mail.polimi.it)).

Monica Nicoli is with the Dipartimento di Ingegneria Gestionale (DIG), Politecnico di Milano, 20133 Milan, Italy (e-mail: [monica.nicoli@polimi.it](mailto:monica.nicoli@polimi.it)).  
Digital Object Identifier 10.1109/TSP.2023.3304002

## I. INTRODUCTION

### A. Contextualization and Background

IN the last two decades, driving automation functionalities have advanced at an incredible rate, allowing an accurate perception of the environment for enhancing vehicle safety [1], [2]. At the same time, the development of cellular communications for the automotive vertical (e.g., 5G and beyond) is driving a new connectivity paradigm for mobility [3], [4], [5]. Vehicle-to-Everything (V2X) communications enable a seamless information sharing among vehicles, road infrastructures and any other road entity over Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) links. Examples of exchanged information in V2X networks include sensor data, driving intents and planned trajectories, or safety-related messages [6]. Moreover, V2X communications allow to extend the ego-sensing capabilities beyond the immediate field of view of on-board sensors, enabling the cooperation across sensing systems of different vehicles. The aggregation of data from spatially distributed sensors (both on vehicles and road infrastructure) through V2X links fosters the deployment of the so called Cooperative Localization (CL) systems [7], [8], [9], [10], [11], [12]. A relevant use case for CL in V2X networks is related to Vulnerable Road User (VRU) detection [13], [14], where cooperation can significantly improve the detection capability.

Regardless of whether these sensors are located in the same vehicle [15], [16] or across different units [17], [18], [19], CL heavily relies on the correct association of sensor measurements, i.e., data association [20], [21], [22]. While data association may appear as a simple task, numerous studies have emphasized the importance of addressing this problem due to the limitations of naive solutions that simply associate closely detected objects [23], [24], [25], [26]. These solutions only yield meaningful results if all vehicles detect an identical number of objects, which is an unrealistic assumption due to varying sensor hardware and fields of view, and do not have false alarms due to clutter. Consequently, it is essential to associate multiple sets of measurements related to distinct detected objects that are only partially in common among vehicles.

In the literature, classical approaches for data association were developed for solving the Multiple Object Tracking (MOT) problem, with the ultimate goal of estimating the trajectories of unknown and time-varying objects. Differently

from the localization of active devices, passive targets produce unknown measurement-to-target connections, which have to be associated before running any CL algorithm. The fusion of multiple sensors' measurements can take place under centralized [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], distributed [37], [38], [39], [40], [41], [42] or hybrid network architectures [43], [44].

In this paper, we focus on a centralized solution in V2X networks where a central processing unit is in charge of combining the raw data (or derived characteristics) from all connected vehicles. Specifically, the aggregated measurements refer to bounding boxes extracted from lidar sensors at the vehicles. Centralized solutions in the literature mainly rely on probability-based methods such as Belief Propagation (BP), also known as Sum-Product Algorithm (SPA), which gives a systematic approximation of optimal Bayesian inference with an appealing performance-complexity trade-off [45]. BP uses an iterative message passing exchange of information over a suitable graph characterizing the specific problem. BP-based techniques are optimal in case of linear and Gaussian models, but provide only an approximation in case of loopy graphs or statistical distributions arising from real systems [21], [22]. Differently from BP-based solutions, we here propose to use Message Passing Neural Networks (MPNNs), that allow to improve performances upon SPA by directly learning the correct associations and noise distributions from data.

### B. Related Works

First works of MOT task were developed in the domain of radar and sonar tracking [30], [31], [32], [33], [34]. Traditional MOT methods, such as Joint Probabilistic Data Association (JPDA) [46], Linear Joint Integrated PDA (LJIPDA) [47] and Multiple Hypothesis Tracker (MHT) [48], assume that the number of targets is known and jointly estimate the target states and association variables. These approaches have been later extended to consider also multi-sensor scenarios [49], [50] as in Linear Multitarget IPDA (LMIPDA) [51]. Recent studies, including probability hypothesis density (PHD) filters [52], [53], adopt finite set statistics to predict the number of targets and target states without directly estimating the association variables. Other studies addressing probabilistic data association can be found in [54], [55], [56], [57], [58], [59]. However, most MOT approaches have limited scalability as the number of sensors and targets grows. Improvements from this point of view have been introduced by BP techniques that are able to achieve high scalability [45]. BP approaches have been investigated for both centralized [27], [28], [29], [36] and distributed [37], [38] solutions.

As far as the data association is concerned, one of the most prevalent approach is to use a graph formulation, which facilitates the description of relationships among multiple measurements on a same set of detected targets. Many solutions that use graphs to solve data association take into account all feasible assignments at the same time, yielding to an NP-hard combinatorial problem [60], [61]. To reduce the complexity, sub-optimal (greedy) methods have been proposed, casting the problem as a linear one and addressing it through minimum-cost [62] and maximum-flow algorithms [63]. These methods,

however, do not guarantee satisfying performance, especially in cluttered and occluded environments [64], [65].

Another possibility to successfully manage data association is to build Machine Learning (ML) models that directly learn from data. ML, and in particular Deep Neural Networks (DNNs), have been embedded inside graphs thanks to the rise in popularity of Graph Neural Networks (GNNs) [66], [67]. GNNs, and more specifically MPNNs, inherit the message passing structure of SPA to produce the desired output from a set of inputs. Indeed, they have been jointly used with the SPA to improve the overall performances by correcting errors created by cycles and model mismatch [68], [69]. Compared to DNNs, MPNNs have fewer parameters but they can still catch the linear and non-linear relationships between input data and output, being at the same time scalable [70]. Moreover, MPNNs have been shown to outperform BP on loopy graphs, provided that enough training data is available [71].

### C. Contribution

To the best of our knowledge, GNNs have never been explored for cooperative sensing nor for vehicular networks. Drawing inspiration from [72], where the detections obtained by a single camera system were associated over consecutive time frames, we here modify and extend the approach to a cooperative (i.e., multi-vehicle) scenario. We consider a centralized network of vehicles, each with a single lidar sensor, with overlapped Field-of-VIEWS (FOVs) allowing a cooperative detection (at the same time instant) of pedestrians through the association of multiple bounding boxes extracted from the lidar point cloud. We selected pedestrians as they are passive elements of the environment and are extremely relevant for safety-related applications (e.g., vulnerable road user protection) as well as they are popularly present in urban areas. Alternatively, vehicles could also be used, but they are typically equipped with active devices in Cooperative Intelligent Transport Systems (C-ITS) (following V2X paradigms), thus notifying their presence in the near surroundings. On the other hand, passive targets, such as pedestrians, traffic signals or poles [73], are not univocally identified and, therefore, data association is needed for their recognition and cooperative detection by multiple sensors.

We assume that the lidar detection system does not incur false detections (i.e., incorrect bounding boxes), which would require a tracking over time to resolve the ambiguity; here we focus on a snapshot-based data association. This assumption may not always hold in real-world scenarios, especially when objects are partially occluded. However, we employ a filtering strategy that is widely adopted in deep learning object tracking and discards unlikely bounding boxes with low detection confidence (see e.g., [74] for a more complete discussion). This helps limiting the false positives as very unlikely bounding boxes are automatically removed by the detector<sup>1</sup>. We also assume the noise statistics as invariant across all sensors, a condition which in practice might not be fulfilled due to different hardware and

<sup>1</sup>Complete removal of false positives may be accomplished by not only taking into account the detection confidence but also the temporal dependencies of detections across adjacent time instants. This extension, not considered here, requires solving the data association problem over multiple graphs relating to adjacent time instants.

lidar processing techniques. Including the variation over time of uncertainties and designing an MPNN-based tracker with data association are non-trivial issues that require deeper research relying on this first activity as a starting point.

We propose an ML approach based on GNNs that exploits the availability of training data, today largely accessible in most applications. In particular, we address the data association problem through MPNNs in a V2X network where vehicles share the detections of lidar sensors in a common infrastructure (e.g., cloud-based). We choose to focus on data association using MPNNs for two main reasons. First, data association is a crucial component in cooperative sensing algorithms, as accurate assignment of detections to tracks significantly impacts the tracking performance: our goal is to provide a solid foundation for more robust tracking solutions. Second, using MPNNs for data association is an innovative approach with great potentials for learning complex relationships in graph-structured data, providing valuable insights and future research opportunities.

A preliminary version of the proposed method has been presented in [75], where we developed a unique graph representation of the data association problem which is handled by an MPNN model that captures the measurements' characteristics and produces a compact and effective feature representation. While in [75] we focused on a simple proof-of-work implementation and stand-alone validation in a vehicular scenario, in this paper we extend the work with the main following contributions:

- proposal of MPNNs models for the cooperative association of 3D bounding boxes from lidar sensing in vehicular networks;
- analysis of the generalization capabilities of the MPNN association model over a number of different and realistic measurement statistics;
- validation of the suggested approach in a realistic cooperative vehicular environment simulated with CARLA [76] where a central unit fuses the bounding boxes obtained by multiple vehicles from on-board lidar data using the PointPillars [77] model;
- comparison with the conventional Sum-Product Algorithm for Data Association (SPADA) [21], [22], with particular focus on association performances and generalization properties.

Note that the assessment on a synthetic cooperative dataset considers the use of an efficient 3D object detector, which has been demonstrated to provide accurate performances in challenging real-world datasets [77]. Since the primary focus of this work is on the fusion over the V2X networks of bounding boxes from multiple vehicles and not on the processing of raw lidar point clouds, any signal losses or adverse weather conditions are not affecting the proposed MPNN as they only reduce the performance of the 3D object detector operating over the lidar point cloud.

Numerical results show that the proposed method is able to efficiently address the data association issue in cooperative connected multi-vehicle systems, and to correctly learn extremely complex (e.g., multi-modal) distributions, such as

the realistic PointPillars outputs. Moreover, with respect to SPADA, the proposed MPNN model can achieve higher performances across different noise statistics and intensities in several circumstances.

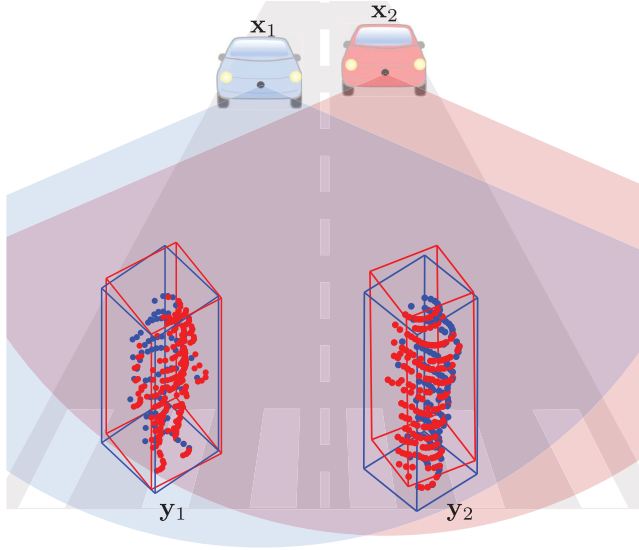
#### D. Paper Organization

This paper is organized as follows: Section II introduces the system model of the cooperative sensing scenario and its graph representation. Section III firstly provides an introduction on the working principle of GNNs, and then defines the proposed MPNN solution. Section IV is devoted to performance analysis in a cooperative vehicular scenario with lidar-based pedestrian detection and to the comparison with SPADA. Finally, Section V draws the conclusion.

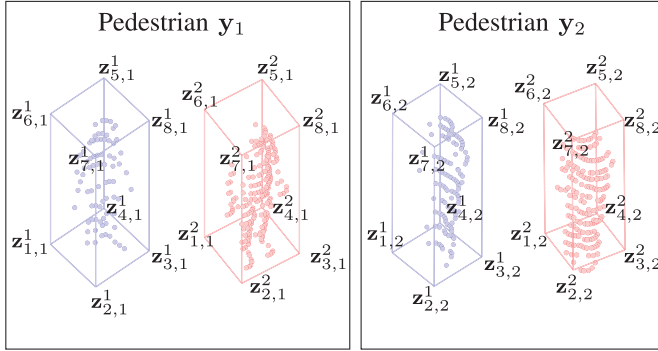
## II. SYSTEM MODEL

Let us denote with  $\mathcal{S}_n = \{1, \dots, S_n\}$  a set of connected vehicles at time  $n$ . A vehicle  $s \in \mathcal{S}_n$  is described by the state vector  $\mathbf{x}_{s,n}$ , which can include kinematic (e.g., position, velocity, etc.) and non-kinematic (e.g., identification number, category, dimension, etc.) parameters. All vehicles are connected to a central processing unit (e.g., a road side unit or a mobile edge cloud) in charge of aggregating the vehicle-generated information and providing a cooperative detection system. We assume an always-available connectivity: model and effects of the communication protocol are out of the scopes of this paper. Each vehicle has a lidar sensing system embedding an ML algorithm for detecting non-cooperative vulnerable road users, here pedestrians, referred to as targets. The  $k$ -th target is described by the state vector  $\mathbf{y}_{k,n}$ , while the set  $\mathcal{Y}_{s,n}$  includes all the pedestrians detectable by vehicle  $s$  (i.e., within its FOV) at time  $n$ . By processing the lidar point clouds gathered at the vehicles via 3D object detection methods, such as [78], [79], each target falling within the lidar sensing range can be recognized and represented by a bounding box encoding its location, extension and rotation. Each target is assumed to generate at most one bounding box at a vehicle per each time step. This assumption, known as “data association assumption” [80], is common in object detection models for lidar point clouds, and more in general in MOT algorithms, as it helps to simplify the detection and tracking process, reduce ambiguities, and improve the overall tracking performance. The  $m$ -th bounding box at vehicle  $s$  at time  $n$  is  $\mathbf{z}_{m,n}^s$  and the associated target is unknown. As such, at time  $n$ , a sensor has a set of unpaired (to the originating target) bounding boxes  $\mathbf{z}_n^s = \{\mathbf{z}_{1,n}^s \dots \mathbf{z}_{M,n}^s\}$ . Note that the set  $\mathbf{z}_n^s$  could even be empty. The union set of all bounding boxes of all vehicles at time  $n$  is  $\mathcal{Z}_n = \bigcup_{s=1}^{S_n} \mathbf{z}_n^s$ .

To visualize the considered vehicular scenario, in Fig. 1(a) we report the case of two vehicles,  $\mathbf{x}_{1,n}$  and  $\mathbf{x}_{2,n}$ , jointly detecting two pedestrians,  $\mathbf{y}_{1,n}$  and  $\mathbf{y}_{2,n}$ , through the bounding boxes  $\mathbf{z}_n^1 = \{\mathbf{z}_{1,n}^1 \mathbf{z}_{2,n}^1\}$  and  $\mathbf{z}_n^2 = \{\mathbf{z}_{1,n}^2 \mathbf{z}_{2,n}^2\}$  for vehicles  $s = 1$  and  $s = 2$ , respectively. The measurement  $\mathbf{z}_{m,n}^s$  is described by the 3D coordinates of its eight corners, i.e.,  $\mathbf{z}_{m,n}^s = [\mathbf{z}_{i,m,n}^s]_{i=1}^8$ , as shown in Fig. 1(b), which take into account the overall footprint and orientation of the target. To correctly associate the bounding



(a) Cooperative detection of pedestrians through lidar sensing



(b) Definition of lidar detections (bounding boxes)

Fig. 1. (a) Cooperative scenario with two vehicles  $\mathbf{x}_1$  and  $\mathbf{x}_2$  detecting pedestrians  $\mathbf{y}_1$  and  $\mathbf{y}_2$  by means of lidar technology. (b) Bounding boxes extracted from the lidar point cloud with corner definition. The subscript  $n$  is removed for visualization purposes.

boxes, an absolute (fixed) Cartesian spatial reference system has to be used for the identification of the corners. In this case, we choose to label as  $\mathbf{z}_{1,m,n}^s$  the bottom-north-east corner and  $\mathbf{z}_{8,m,n}^s$  the top-south-west one.

In the proposed GNN solution, the union set  $\mathcal{Z}_n$  is modeled as a direct graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where each node  $i \in \mathcal{V}$  corresponds to a single measurement, while the edge  $(i, j)$ , with  $i \neq j$ , indicates a candidate association. To univocally map the node  $i \in \mathcal{V}$  with the measurement  $m$  of vehicle  $s$  at time  $n$ , we define the mapping function  $\Phi_n : \mathcal{V} \rightarrow \mathcal{Z}_n \times \mathcal{S}_n$ . The function  $\Phi_n(i) = \{m, s\}$  cannot inherently prevent the association of two distinct measurements of a same vehicle. For this reason, we also introduce the association-related variable  $y_{i \rightarrow j} \in \{0, 1\}$  which denotes the presence/absence of the edge  $(i, j)$ , i.e., the two bounding boxes embodied in nodes  $i$  and  $j$  refer to a same target. The goal of the data association algorithm (here addressed with MPNN) is to estimate the association variable  $\hat{y}_{i \rightarrow j} \in \{0, 1\}$  by considering all possible pairings of bounding boxes, with the constraint of  $\hat{y}_{i \rightarrow j} = 0$  if the mappings  $\Phi_n(i)$  and  $\Phi_n(j)$  refer to a same vehicle  $s$ .

TABLE I  
SUMMARY TABLE OF NOTATION

Description	Symbol
Set of vehicles at time $n$	$\mathcal{S}_n$
State of vehicle $s$ at time $n$	$\mathbf{x}_{s,n}$
Set of detectable targets by vehicle $s$ at time $n$	$\mathcal{Y}_{s,n}$
State of target $k$ at time $n$	$\mathbf{y}_{k,n}$
Set of bounding boxes of all vehicles at time $n$	$\mathcal{Z}_n$
Set of bounding boxes of vehicle $s$ at time $n$	$\mathcal{z}_n^s$
$m$ -th bounding box of vehicle $s$ at time $n$ described by the coordinates of its corners	$\mathbf{z}_{m,n}^s = [\mathbf{z}_{i,m,n}^s]_{i=1}^8$
GNN directed graph with vertex and edges	$\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Mapping function from node $i \in \mathcal{V}$ to measurement $m$ of vehicle $s$ at time $n$	$\Phi_n : \mathcal{V} \rightarrow \mathcal{Z}_n \times \mathcal{S}_n$
Association-related variable	$y_{i \rightarrow j} \in \{0, 1\}$

As an example of graph construction, we refer to the vehicular scenario shown in Fig. 2(a) where vehicle  $\mathbf{x}_{2,n}$  detects only  $\mathbf{y}_{1,n}$  through measurement  $\mathbf{z}_{1,n}^2$ , while the other two vehicles  $\mathbf{x}_{1,n}$  and  $\mathbf{x}_{3,n}$  can detect both targets  $\mathbf{y}_{1,n}$  and  $\mathbf{y}_{2,n}$ , respectively. It follows that the graph with true measurements association for such scenario is the one indicated in Fig. 2(b), which has to be reconstructed from the fully-connected graph in Fig. 2(c) that includes all possible pairings. In next section, we detail the proposed algorithm for estimating the connections from all possible associations, i.e., how to get the graph in Fig. 2(b) from the one in Fig. 2(c).

A summary of the main notation variables introduced in this section and their description is provided in Table I.

### III. ADDRESSING DATA ASSOCIATION WITH MPNN

In this section, we first introduce the general concept of GNN and more specifically MPNN (Section III-A), which is the base for the proposed model. Then, we define the proposed MPNN model with an insight on possible classification strategies. Finally, we describe the loss function used to train the model, as well as the performance metrics.

#### A. Introduction to GNNs

Neural networks acting on graphs have been investigated for more than a decade, being originally referred to as GNNs [66], [67] and successively extended to many variants such as MPNNs [81]. A complete generalization of GNNs is formulated in [82] under the name of Graph Networks (GNs). Models in this ML family have been studied in supervised, semi-supervised, unsupervised, and reinforcement learning contexts across a wide range of problem domains. They have been used to learn the dynamics of physical systems [83], predict the chemical properties of molecules [84], optimize the communication in multi-agent networks [85], or even employed in machine translation [86]. A further domain of applications includes vehicular environments, where GNN are used to predict road traffic [87], [88] or classify and segment 3D meshes and point clouds [89].

We here consider an MPNN that iteratively performs a message passing procedure over a graph  $\mathcal{G}$ . Iterations are indexed

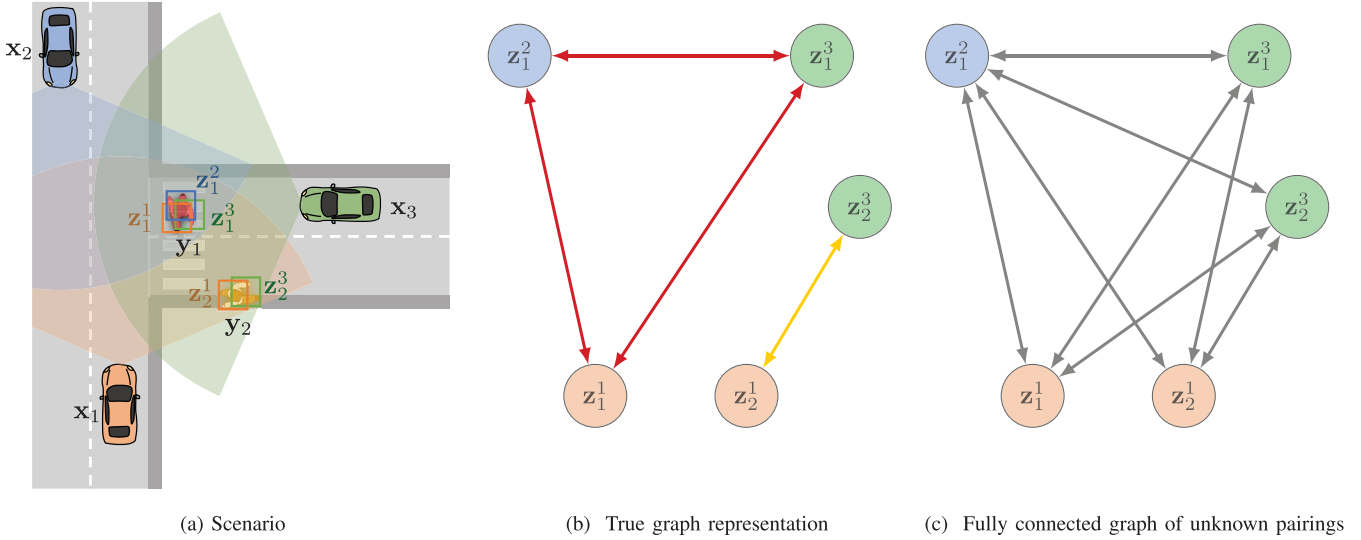


Fig. 2. (a) Top view of an exemplary cooperative localization scenario with three vehicles detecting one or two pedestrians each. (b) Virtual graph representing the pairings measurement at multiple vehicles. Each subgraph includes all measurements of a same target. (c) Graph with unknown associations among all measurements. Each edge embodies a potential association, which has to be probabilistically computed to get the result in (b). The color of nodes refers to the color of originating vehicle, while colors of edge indicate the two detected pedestrians.

with  $t$ , the maximum number of message passing steps (a design parameter) as  $T$ , while  $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$  is the set of neighbors of node  $i \in \mathcal{V}$ . We also identify the so called embeddings, i.e., attributes, of node  $i$  and edge  $(i, j)$  with variables  $\mathbf{h}_i^{(t)}$  and  $\mathbf{m}_{i \rightarrow j}^{(t)}$ , respectively.

The purpose of the MPNN is to train a function that propagates information from node and edge embeddings/attributes throughout  $\mathcal{G}$ . The more message passing steps are performed, the more the node and edge embeddings contain elaborated information, just like the receptive field of a Convolutional Neural Network (CNN). To this extent, a Neural Network (NN) is present at each node and edge of the graph. The NN at node is indicated with  $g_n(\cdot)$ , while the one over the edge by  $g_e(\cdot)$ . Considering that  $g_n(\cdot)$  and  $g_e(\cdot)$  have the same parameters, respectively across each node and each edge, they may be trained on small-scale graphs before being applied to large-scale problems.

For each iteration  $t = 1, \dots, T$ , each node  $i \in \mathcal{V}$  sends the following message to its neighbors  $\mathcal{N}_i$

$$\mathbf{m}_{i \rightarrow j}^{(t)} = g_e(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{m}_{i \rightarrow j}^{(t-1)}), \quad \forall j \in \mathcal{N}_i, \quad (1)$$

with

$$\mathbf{h}_i^{(t)} = g_n(\mathbf{h}_i^{(t-1)}, \Phi(\{\mathbf{m}_{j \rightarrow i}^{(t)}\}_{j \in \mathcal{N}_i})). \quad (2)$$

Function  $\Phi(\cdot)$  is called *aggregation* function and it is invariant to permutations of its inputs (e.g., element-wise summation, mean, maximum). Concisely, and referring to Fig. 3, the message  $\mathbf{m}_{i \rightarrow j}^{(t)}$  sent from node  $i$  over an edge  $(i, j)$  updates the previously sent message  $\mathbf{m}_{i \rightarrow j}^{(t-1)}$  over the same edge with the available attributes  $\mathbf{h}_i^{(t-1)}$  and  $\mathbf{h}_j^{(t-1)}$  of the involved nodes, respectively. The attribute  $\mathbf{h}_i^{(t)}$  is obtained by combining together all the incoming messages at the node  $i$ , i.e.,  $\mathbf{m}_{j \rightarrow i}^{(t)} \forall j \in \mathcal{N}_i$  (through function  $\Phi$ ) and the previously available information  $\mathbf{h}_i^{(t-1)}$  (computed at previous iteration).

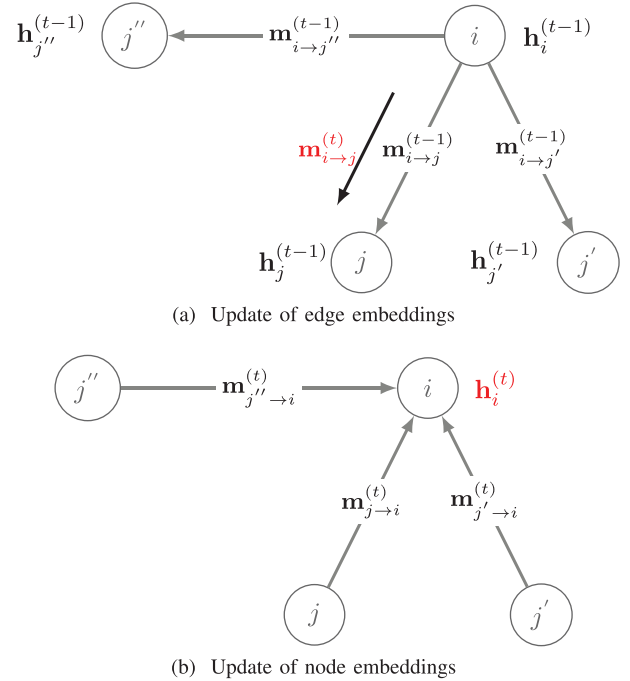


Fig. 3. MPNN working principle: (a) update of edge embeddings, (b) update of node embeddings. The updated elements are in red. For each MPNN step  $t$ , first compute the edge embeddings from node  $i$  over all edges (i.e., toward all neighbors  $\mathcal{N}_i$ ) according to (1). Then, compute the node embeddings depending on the previously updated edge embeddings as in (2).

## B. MPNN Model for Data Association

The proposed model consists of two parts: an MPNN and an edge classifier. The role of the MPNN is to process the input graph  $\mathcal{G}$  derived from the measurements of all vehicles at a given time  $n$ , i.e.,  $\mathcal{Z}_n$ . On the other hand, the edge classifier is a binary classifier with the role of determining the pairings of all the measurements referring to the same target, i.e., finding the association variable  $\hat{y}_{i \rightarrow j}$  based on association probabilities.

As a consequence, at the output of the classifier we have a set of multiple disjoint subgraphs (as in Fig. 2(b)), each of them grouping all the measurements that are hypothesized to be originated from the same target.

The MPNN model is composed of four multi-layer perceptrons (MLPs):  $g_e(\cdot)$  at each edge and  $g_n^{\text{in}}(\cdot)$ ,  $g_n^{\text{out}}(\cdot)$ ,  $g_n(\cdot)$  at each node. The role of MLPs  $g_e(\cdot)$  and  $g_n(\cdot)$  is to update the edge and node embeddings, respectively, in a similar way as the conventional MPNN in (1) and (2). On the other side,  $g_n^{\text{in}}(\cdot)$  and  $g_n^{\text{out}}(\cdot)$  are introduced to better encode the structure of incoming and outgoing edges. In this way, we can split the problem into two parts and individually manage the incoming and outgoing edges in each node.

The message passing over the graph works as follows. First, we update the edges embeddings as in (1), while the node embeddings are updated taking into account both the incoming  $\mathbf{m}_{j \rightarrow i}^{\text{in},(t)}$  and outgoing  $\mathbf{m}_{i \rightarrow j}^{\text{out},(t)}$  edge embeddings as

$$\mathbf{h}_i^{(t)} = g_n \left( \sum_{j \in \mathcal{N}_i} \mathbf{m}_{j \rightarrow i}^{\text{in},(t)}, \sum_{j \in \mathcal{N}_i} \mathbf{m}_{i \rightarrow j}^{\text{out},(t)} \right), \quad (3)$$

where  $\mathbf{m}_{j \rightarrow i}^{\text{in},(t)}$  and  $\mathbf{m}_{i \rightarrow j}^{\text{out},(t)}$  are defined as

$$\mathbf{m}_{j \rightarrow i}^{\text{in},(t)} = g_n^{\text{in}} \left( \mathbf{h}_i^{(t-1)}, \mathbf{m}_{j \rightarrow i}^{(t)} \right), \quad \forall j \in \mathcal{N}_i, \quad (4)$$

$$\mathbf{m}_{i \rightarrow j}^{\text{out},(t)} = g_n^{\text{out}} \left( \mathbf{h}_i^{(t-1)}, \mathbf{m}_{i \rightarrow j}^{(t)} \right), \quad \forall j \in \mathcal{N}_i. \quad (5)$$

We remark that this is done to divide the problem into two parts, as the constructed graph for solving the data association is bi-directed (i.e., undirected), which is common in most graphs used by MPNNs. However, our approach also needs to ensure the unique constraints of our data association problem, i.e., that the association-edge between two measurements is conceptually the same in both directions.

After  $T$  message passing steps, the edge embeddings  $\mathbf{m}_{i \rightarrow j}^{(T)}$  are fed into an MLP edge classifier  $g_e^{\text{class}}(\cdot)$  which evaluates the association probabilities  $\hat{y}_{i \rightarrow j}^{(T)}$  as

$$\hat{y}_{i \rightarrow j}^{(T)} = g_e^{\text{class}} \left( \mathbf{m}_{i \rightarrow j}^{(T)} \right), \quad \forall (i, j) \in \mathcal{E}. \quad (6)$$

The association variables  $\hat{y}_{i \rightarrow j}^{(T)}$  are then obtained with a thresholding operation, with threshold  $\Gamma$ , to pair nodes  $i$  and  $j$ . Two nodes are associated (i.e., two bounding boxes at distinct vehicles refer to a same pedestrian) if

$$\hat{y}_{i \rightarrow j}^{(T)} \geq \Gamma, \quad (7)$$

which implies  $\hat{y}_{i \rightarrow j} \triangleq 1$ . However, it may happen that one measurement of a vehicle is associated to multiple measurements of another vehicle. To avoid this issue, a constraint is enforced such that a bounding box of a vehicle can be associated to *at most* one bounding box of another vehicle.

### C. Loss and Performance Metrics

For computing the training loss and performing back-propagation, we employ the weighted binary cross-entropy that

is estimated at the end of each message passing iteration  $t$  after the edge classifier's prediction  $\hat{y}_{i \rightarrow j}^{(t)}$  as

$$\mathcal{L} = -\frac{1}{|\mathcal{E}|} \sum_{t=1}^T \sum_{(i,j) \in \mathcal{E}} \left\{ (1-y_{i \rightarrow j}) \log \left( 1 - \hat{y}_{i \rightarrow j}^{(t)} \right) + w y_{i \rightarrow j} \log \left( \hat{y}_{i \rightarrow j}^{(t)} \right) \right\}, \quad (8)$$

where  $w$  is a weight given to the positive class in order to compensate the class unbalances and it is computed as

$$w = \frac{\sum_{(i,j) \in \mathcal{E}} \mathbf{1}(y_{i \rightarrow j} = 0)}{\sum_{(i,j) \in \mathcal{E}} \mathbf{1}(y_{i \rightarrow j} = 1)}, \quad (9)$$

where  $\mathbf{1}(\cdot)$  is an indicator function that returns 1 if the condition is true and 0 otherwise. Concerning the performance metrics, we adopt the accuracy measure defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \quad (10)$$

where the terms TP, TN, FP and FN indicate the number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), respectively.

## IV. SIMULATION EXPERIMENTS

To evaluate the proposed MPNN model for data association we consider a network of vehicles localizing pedestrians through lidar sensing. We dedicate Section IV-A to the simulation scenario and dataset, while Section IV-B reports the results of performed simulations.

### A. Simulation Scenario

Due to the unavailability of real-world cooperative perception datasets, i.e., collected by multiple and synchronous lidar-equipped vehicles, we here employ a simulator of automated driving systems that allows us to generate lidar readings at multiple vehicles moving in a synthetic, yet realistic, mobility environment. Similarly to [90], we use the CARLA simulator [76], an extremely advanced software that integrates trajectory planning and sensing. The considered scenario is referred to as *Town02* in the simulator, which spans over an area of roughly 200 m  $\times$  200 m. Twenty vehicles with lidar and fifty pedestrians populate the scene, unless otherwise specified. The state  $\mathbf{x}_{s,n}$  of each vehicle refers to its 3D position. A snapshot restricted to seven vehicles with associated point clouds of the simulator is show in Fig. 4, where we represent the effect of cooperative sensing by merging seven lidar point clouds. Specifically, for visualization purposes we group the vehicles into three subgroups and we show the partial point cloud in Figs. 4(a), 4(b) and 4(c), respectively, while the cooperative perception obtained by merging all the seven point clouds is in Fig. 4(d).

The duration of simulation is 300 s, with sampling time of 0.2 s. This results in 1500 snapshots of the scene, each one described by vehicles and pedestrians' positions and lidar detections. A top-view image of the simulation in a fixed time instant is shown in Fig. 5, where we include both vehicles (red squares) and detected pedestrians (blue triangles) as well as the

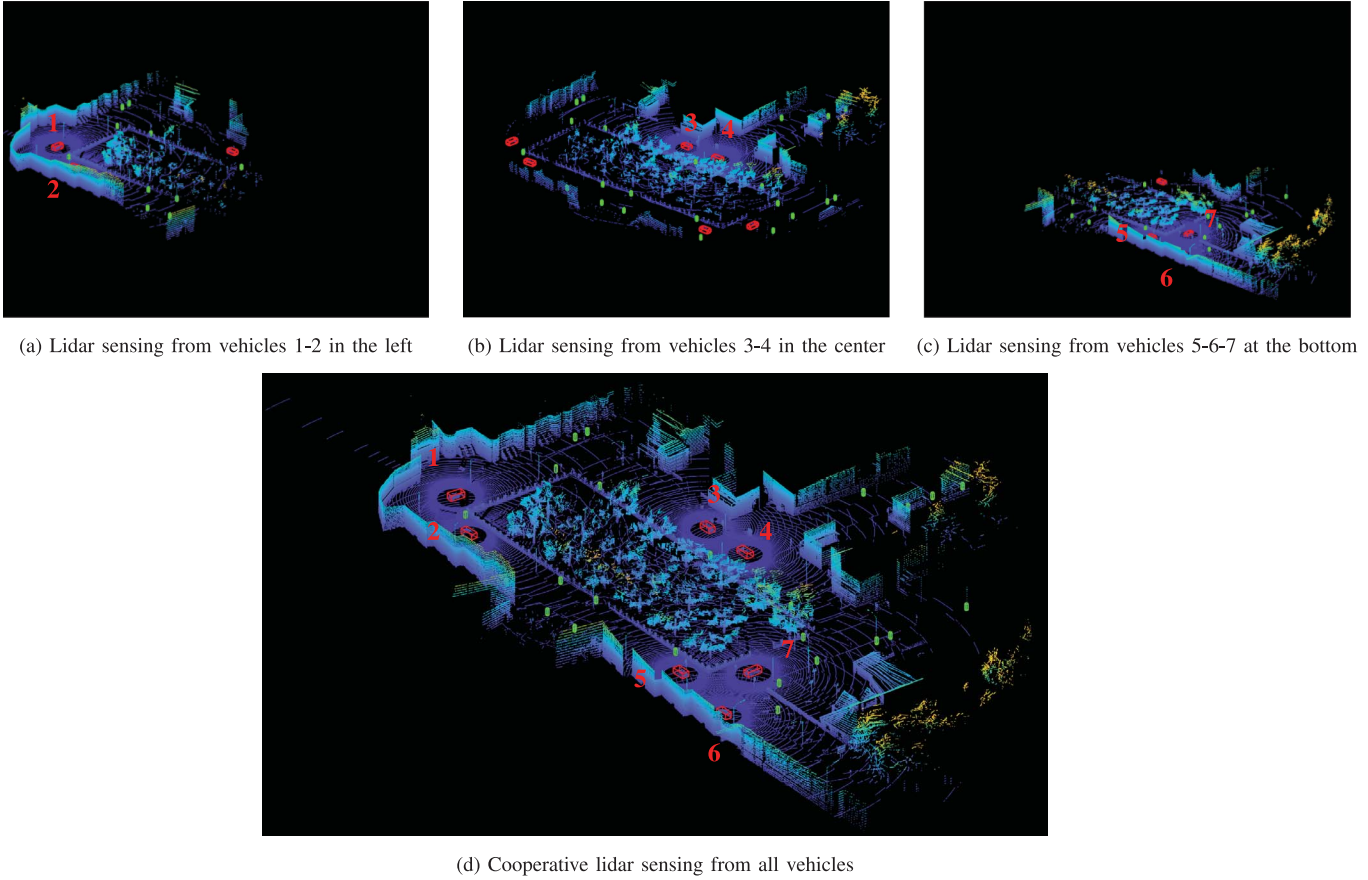


Fig. 4. Snapshot of lidar sensing simulated by CARLA: seven vehicles (red bounding boxes) detect pedestrians (green bounding boxes). (a) Partial lidar sensing from the two vehicles in the left. (b) Partial lidar sensing from the two vehicles in the center. (c) Partial lidar sensing from three vehicles at the bottom. (d) Cooperative sensing as a combination of seven lidar point clouds.

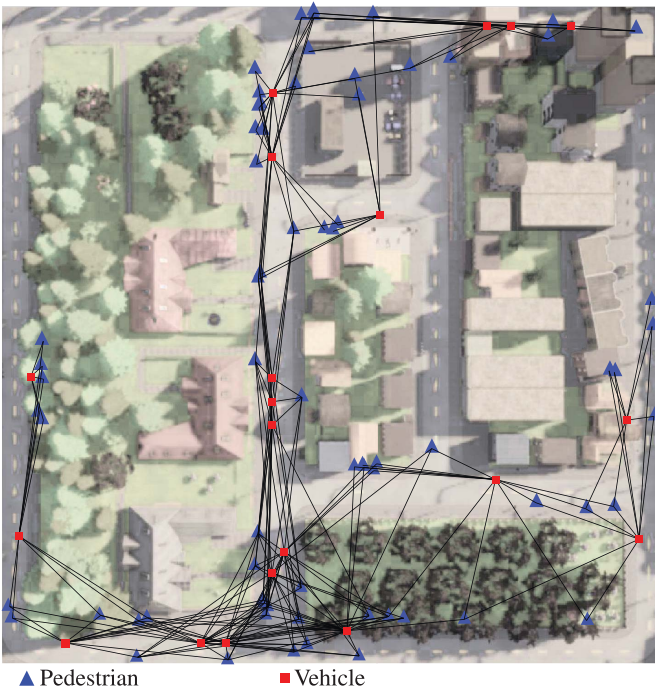


Fig. 5. Top-view of the simulation scenario with twenty vehicles (red squares), detected pedestrians (blue triangles) and detections (black lines).

associated detections (black link). A vehicle  $s \in \mathcal{S}$  can detect a pedestrian  $k \in \mathcal{F}_{v,n}$  if it falls in its field of view. All lidar sensors are configured to run at a 5 Hz update frequency, with an FOV of 360 deg in azimuth and  $[-30, +10]$  deg in elevation. The number of channels supported is 64, corresponding to a spatial resolution of 0.625 deg. The sensing range is limited to 70 m and the number of points of the cloud cannot exceed 1 million per second. The single point has an accuracy of  $\pm 2$  cm. To simulate realistic operating conditions, 20% of the points are randomly dropped during every simulation frame.

The ground truth information provided by the simulator includes the true positions of vehicles, i.e.,  $\mathbf{x}_{s,n}, \forall s \in \mathcal{S}_n$ , and the true bounding boxes around the pedestrians, defined by its eight corners, i.e.,  $\mathbf{y}_{k,n} = [\mathbf{y}_{i,k,n}]_{i=1}^8$ . Localization errors are introduced as an additive measurement error  $\mathbf{w}_{i,m,n}^s$ , which directly translates over the 3D corners of the measured bounding boxes. The resulting noisy measurement of a bounding box corner is thus defined as:

$$\mathbf{z}_{i,m,n}^s = \mathbf{y}_{i,k,n} + \mathbf{w}_{i,m,n}^s, \quad \forall k \in \mathcal{Y}_{s,n}, \forall i \in \{1, \dots, 8\}. \quad (11)$$

Note that the measurement error distribution is the same for all corners, for all time instants and across all vehicles. Unless otherwise specified, the artificial noise  $\mathbf{w}_{i,m,n}^s$  follows an isotropic

Gaussian distribution with standard deviation  $\sigma = 10$  cm. Note that the additive noise is absent in case of using a ML model for the automatic extraction of bounding boxes, as in the case of PointPillars [77], since the error is embedded in the model itself.

In the simulations, we validate the systems for a variety of noise intensities ranging from extremely accurate detections up to inefficient systems (errors in the order of meters) for the considered vehicular context targeting the automation of mobility. The former case can be considered as a condition in which the vehicle position is assumed to be perfectly known and the only source of error is attributed to lidar sensing and bounding boxes extraction algorithm. The latter case, instead, embeds both vehicle uncertainty and the errors in the generation of bounding boxes. We do not consider separate effects as we aim to assess the aggregated model robustness.

We divide the overall dataset into training (700 samples) and validation (800 samples) parts, with dimensions optimized as discussed in Section IV.B.2. Moreover, in order to assess the generalization of the method, we increase the number of validation samples by applying a random flip along the  $x$  and  $y$  axes of the bounding box positions, thus obtaining a total of 1600 validation samples. We remark that a sample is a snapshot of the scene at a given time instant  $n$  and it is fully represented by the graph of unknown measurement-pairings (Fig. 2(c)). To avoid the computational burden of dealing with too many edges, we introduce a gating which a-priori discards unlikely associations, i.e., ignoring edges related to centroids whose distance is greater than 10 m. As optimizer, we use the Adam optimizer with tuned learning rate of  $10^{-3}$  and hyper-parameters  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  [91]. The performance metrics are computed using the thresholding in (7) with  $\Gamma = 0.5$ .

## B. Simulation Results

1) *Initialization of Node and Edge Embeddings:* To initialize the node and edge embeddings, we adopt a strategy that learns how to extract feature embeddings directly from measurements. This is done by using an MLP at each node and edge, called  $g_n^{\text{enc}}(\cdot)$  and  $g_e^{\text{enc}}(\cdot)$ , respectively. For the considered cooperative lidar sensing scenario, we use the geometric characteristics of the bounding boxes as input to the two neural networks to obtain  $\mathbf{m}_{i \rightarrow j}^{(0)}$  and  $\mathbf{h}_i^{(0)}$  as

$$\mathbf{m}_{i \rightarrow j}^{(0)} = g_e^{\text{enc}} \left( \mathbf{z}_{1,m,n}^s - \mathbf{z}_{1,m',n}^{s'}, \mathbf{z}_{8,m,n}^s - \mathbf{z}_{8,m',n}^{s'} \right),$$

$$\forall (i, j) \in \mathcal{E} : \Phi_n(i) = \{m, s\} \wedge \Phi_n(j) = \{m', s'\}, s \neq s', \quad (12)$$

$$\mathbf{h}_i^{(0)} = g_n^{\text{enc}} \left( \mathbf{z}_{m,n}^s \right) \quad \forall i \in \mathcal{V} : \Phi_n(i) = \{m, s\}. \quad (13)$$

This allows the MPNN to discriminate not only the position of the detected object, but also its dimension and rotation. We find this approach to be highly effective and efficient, as it uses a minimal amount of information for data association, limiting the data exchange among vehicles.

Incorporating additional features, such as individual point cloud positions, into the current feature encoding could be

TABLE II  
IMPACT OF TRAINING DATASET SIZE (NUMBER OF SAMPLES) ON ACCURACY, PRECISION AND RECALL METRICS

# of samples	Accuracy	Precision	Recall
16	0.869	0.852	0.859
32	0.919	0.891	0.912
64	0.972	0.953	0.980
128	0.993	0.989	0.992
300	0.999	0.998	0.998
700	0.999	0.999	0.999

beneficial and would require only modifying the encoding neural networks  $g_n^{\text{enc}}(\cdot)$  and  $g_e^{\text{enc}}(\cdot)$ . While this could potentially enhance performance, there are two primary drawbacks to consider. First, the volume of information that would need to be exchanged with the central entity responsible for data association via MPNN could become unmanageable and unsustainable, given that a lidar sensor typically outputs more than 1 million point clouds per second. Second, increasing the number of input features might inadvertently introduce unrelated or redundant features that may not be beneficial or could even negatively impact the inference process due to the multi-dimensionality problem in machine learning.

2) *Impact of Training Dataset Dimension:* We first analyze the impact of the training dataset size on the model's performance. This is crucial in determining whether the model exhibits high or low bias. In essence, expanding the dataset size decreases the model's variance, meaning the residual error is predominantly due to bias. In Table II, we present the validation accuracy, precision, and recall after 100 epochs for varying training dataset size. We note that by increasing the number of samples, the model improves the performance metrics and reaches an upper bound on the accuracy after 700 samples, representing the best accuracy reachable by the model, i.e., its bias. It is noteworthy that the recall typically overcomes the precision, implying a larger number of false positives than false negatives. This is because the ground-truth graph retains a predominant number of zeroed edges, thus the model is more prone to mistake on edges that are labeled as zeros, despite the loss function employed (8) for unbalanced classes.

3) *Impact of MPNN Iterations:* This assessment aims to verify the role of the number of message passing iterations  $T$ , a fundamental parameter to tune the amount of information extracted and elaborated from the data. In Fig. 6, we show the accuracy (and associated confidence) metric in the validation dataset over the number of epochs for  $T = \{1, 2, 4, 8, 12\}$ . We notice that increasing  $T$  leads to a higher accuracy and a faster convergence, at the cost of increasing computational complexity. However, a saturation condition occurs for  $T > 4$ , leading us to select  $T = 4$  as a good trade-off between accuracy, convergence and complexity. This value will be used for the following analyses.

4) *Impact of the Measurement Statistics:* This assessment has the goal of verifying how the MPNN model handles unobserved noises (for which it has never been trained on). This is extremely useful in case the model is trained in a



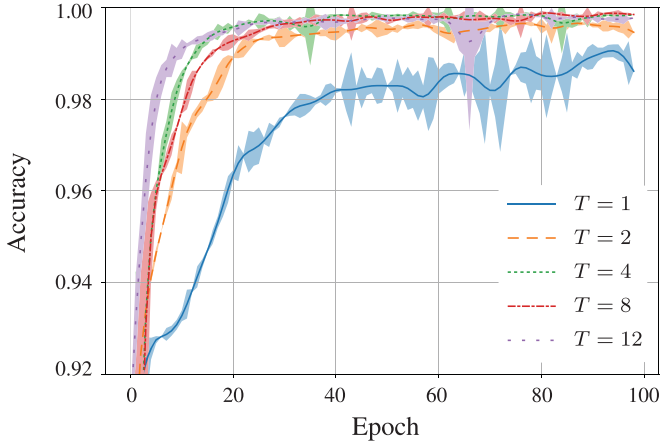


Fig. 6. Accuracy of the MPNN for different values of message passing iterations  $T = [1, 2, 4, 8, 12]$  over the epochs for the validation dataset. The mean value (solid line) is plotted together with the associated uncertainty (shaded area) computed using the maximum and minimum values of accuracy as boundaries.

TABLE III  
DEFINITION OF MEASUREMENT NOISE DISTRIBUTIONS

Distribution	$f_{\mathbf{w}}(\mathbf{w})$
Isotropic Gaussian	$(\sqrt{2\pi\sigma^2})^{-1} \exp(-0.5 \sigma^{-2} \mathbf{w}^2)$
Non-isotropic Gaussian	$(\sqrt{\det(2\pi\Sigma)})^{-1} \exp(-0.5 \mathbf{w}^T \Sigma^{-1} \mathbf{w})$
Laplace	$(\sqrt{2}\sigma)^{-1} \exp(-\sqrt{2} \sigma^{-1}  \mathbf{w} )$
Uniform	$\begin{cases} 0.75 (\pi\sigma^3)^{-1}, & \text{if } \ \mathbf{w}\ _2 < \sigma, \\ 0, & \text{otherwise.} \end{cases}$
Discrete	$\begin{cases} 0.4, & \text{if } \mathbf{w} = \mathbf{0}_3, \\ 0.1, & \text{if } \mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}, \\ 0, & \text{otherwise.} \end{cases}$

simulated and controlled environment, and then deployed in real systems typically characterized by different measurement statistics. Since there is almost no literature detailing the error characteristics of real ML-based 3D object detectors, we investigate the types of noise that are currently considered in point cloud denoising algorithms. As suggested in [92], we explore five different noise statistics: the already introduced isotropic Gaussian, the non-isotropic Gaussian, the Laplace, the uniform, and the discrete one, which are defined by the distributions  $f_{\mathbf{w}}(\mathbf{w}) \triangleq f_{\mathbf{w}}(\mathbf{w}_{i,m,n}^s)$  as reported in Table III, where  $\sigma$  denotes the standard deviation,  $\Sigma = \sigma^2 [[1, -0.5, -0.25]^T, [-0.5, 1, -0.25]^T, [-0.25, -0.25, 1]^T]^T$ , while  $\mathbf{w}_1 = (\pm\sigma, 0, 0)$ ,  $\mathbf{w}_2 = (0, \pm\sigma, 0)$  and  $\mathbf{w}_3 = (0, 0, \pm\sigma)$ .

We also implement the ML model PointPillars [77] to process the lidar point clouds and derive the associated bounding boxes. This method allows us to assess the performance of the MPNN without resorting to artificially generated measurements in (11), leading to a detection system that closely resembles practical scenarios. By using the detections produced by PointPillars as inputs to our data association system, we maintain a noise distribution that mirrors realistic conditions, which is essential for evaluating the effectiveness of the proposed MPNN-based data association strategy in various real-world situations. The

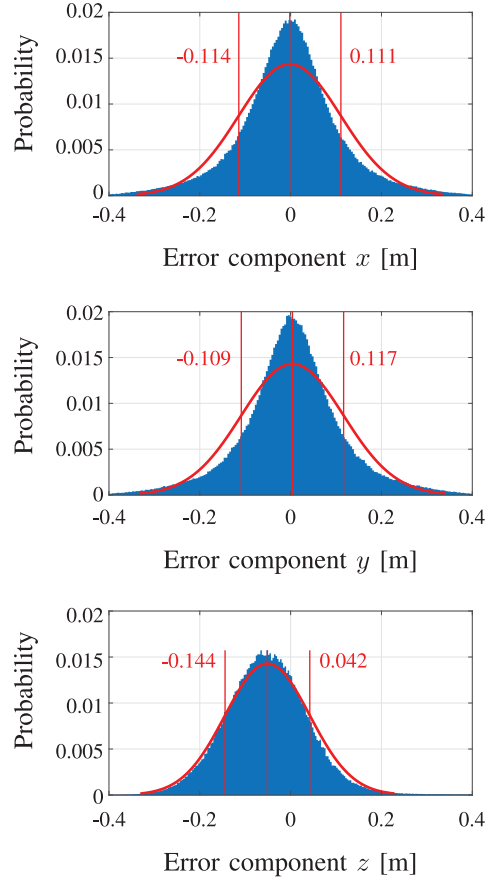


Fig. 7. PointPillars localization error on the corners of bounding boxes along  $x$ ,  $y$  and  $z$  components. The three histograms are approximated with a Gaussian distribution whose standard deviation is highlighted in red.

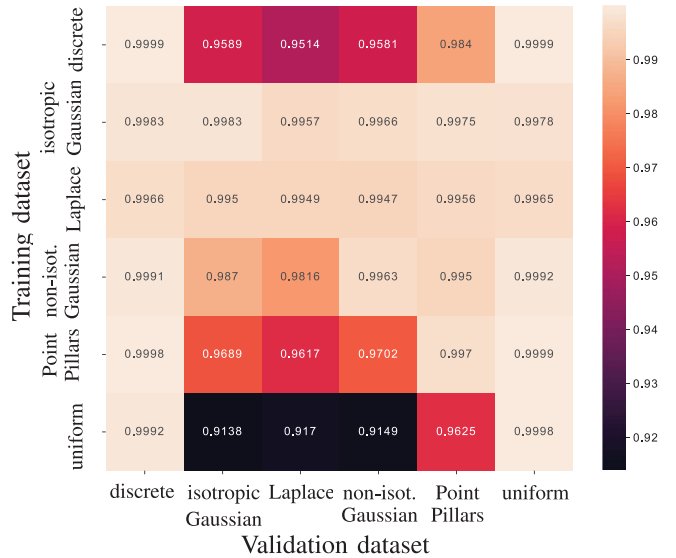


Fig. 8. Accuracy of the MPNN model (after 25 epochs) for training/validation mismatch on measurement noise.

statistics of the 3D localization error (computed as the difference between the true and estimated bounding boxes over  $x$ ,  $y$  and  $z$  axes) of PointPillars are reported in Fig. 7, showing that they can be well approximated by a zero-mean Gaussian

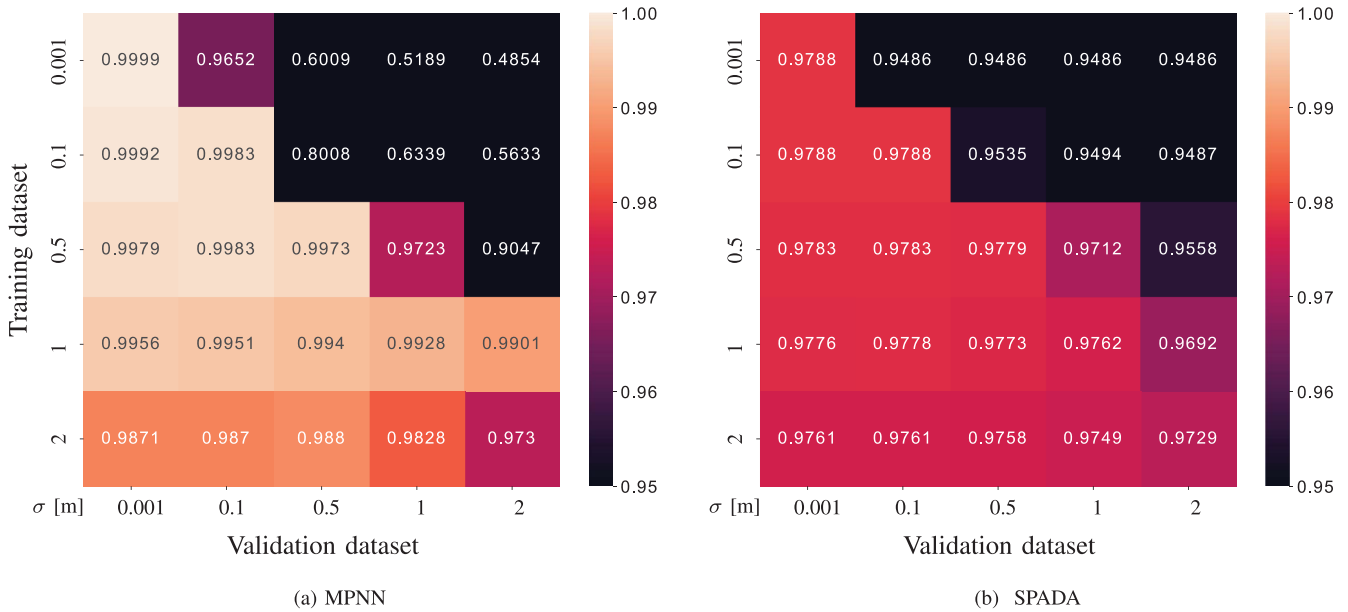


Fig. 9. Comparison on data association accuracy between (a) MPNN and (b) SPADA. For MPNN we show the accuracy (after 25 epochs) for training/validation mismatch on measurement noise, whereas for SPADA we validate each standard deviation of the likelihood with different noise intensities.

distribution over the horizontal  $(x, y)$  space. On the other hand, for the vertical dimension  $z$ , the model is more likely to predict boxes in higher positions (i.e., above the road) instead of the opposite, so the error distribution is slightly biased. Furthermore, we notice that the error statistics do not vary significantly over the three axes, suggesting that they (almost) follow an isotropic Gaussian distribution with standard deviation  $\sigma = 10$  cm.

In Fig. 8 we analyze the mixed impact of training and testing with different noise statistics. The value of  $\sigma$  for each noise distribution has been set to 10 cm as closely matching the standard deviation used for fitting the error statistics of PointPillars. Analyzing the results, it is apparent that training and validating the MPNN model on the same noise distributions lead to optimal performances. This shows that the model is able to obtain good accuracy regardless of the noise type, provided that the same noise is experienced for both training and validation phases. Focusing now on the different combinations of training/validation noises, results detail that training under the isotropic Gaussian or Laplace noise allows the model to generalize well over all noise types, suggesting that these distributions may be employed in real-life applications where noise statistics are not known beforehand. On the other hand, training considering discrete and/or uniform leads to poor generalization results during validation, most probably due to the simplistic noise distributions compared to all other noise types. Finally, training on realistic data, i.e., over the noise generated by PointPillars, does not allow the MPNN to generalize well over other distributions, particularly for Laplace and isotropic/non-isotropic Gaussian noises.

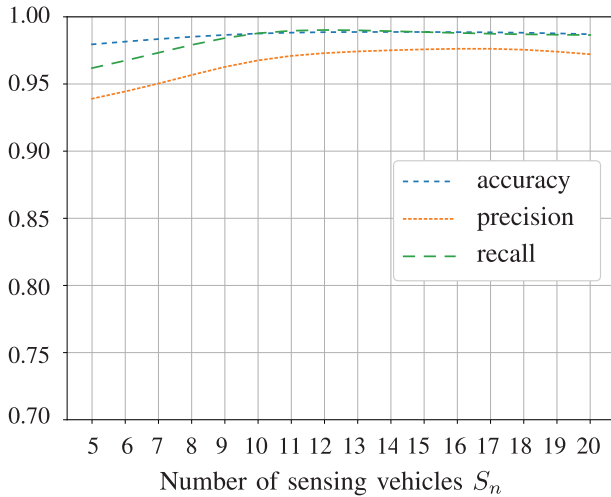
5) *Impact of the Different Scenarios and False Positives:* This experiment aims at verifying the validation performances of the proposed model in a brand new scenario where false positives, i.e., false alarms, are present. This allows us to assess the robustness and adaptability of the model in more

realistic conditions, demonstrating its potential for practical implementation.

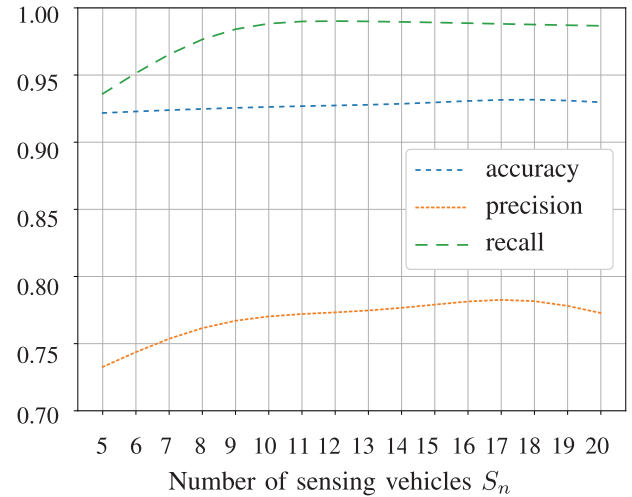
To this purpose, in Fig. 10, we report the results of performance validation in *Town10* scenario of CARLA simulator, where we vary the number of cooperating vehicles from 5 to 20. The proposed MPNN association strategy is evaluated considering both the absence (Fig. 10(a)) and presence (Fig. 10(b)) of false alarms, which are obtained from the Pointpillars detector. We would like to highlight that the model has been trained in the map *Town2* illustrated in Fig. 5 neglecting any false positive, thus *Town10* and the presence of false alarms are unseen conditions. Starting from the scenario without false alarms in Fig. 10(a), we note that increasing the number of vehicles leads to better accuracy, up to a plateau around 97%, which is just 2% below the results in the original scenario with Pointpillars (see Fig. 8). Accounting for the false positives, we notice in Fig. 10(b) a decrease of the accuracy to 93%. Even more relevant is the precision which falls to 78% due to the fact that each false positive introduces new nodes and edges in the graph which will be associated with real detections, leading to lower performances.

6) *MPNN vs SPADA - Generalization Capabilities:* This experiment compares the performance of the proposed MPNN association model against a conventional SPADA over different combinations of Gaussian noise intensities used in the training and validation datasets. For the SPADA, a training phase is not needed, but we can embed prior knowledge on the noise intensity by calibrating the standard deviation used for computing the measurement likelihood function. To do so, we process the training dataset and extract a single standard deviation value that characterizes the considered noise intensity.

The comparison is reported in Fig. 9. Regarding the MPNN, we show, for different training and validation datasets, the validation accuracy reached after 25 epochs, while for the



(a) Unseen scenario without false alarms



(b) Unseen scenario with false alarms

Fig. 10. Analysis of validation accuracy, precision and recall in an unseen scenario, for different number of cooperative vehicles: (a) absence of false alarms, (b) presence of false alarms.

SPADA we represent the validation accuracy after convergence using different a-priori noise statistics (in terms of standard deviation). First, we can clearly observe that, in both algorithms, the bottom-left part of the matrix has higher values of accuracy if compared with the top-right part. This is due to the fact that, generally, overestimating the noise (i.e., bottom-left part) leads to a more robust model that can handle noises with lower intensity. On the contrary, underestimating the noise (i.e., top-right part) can incur into problematic situations, especially in the case of MPNN (Fig. 9(a)). From this point of view, SPADA (Fig. 9(b)) is more solid and can better handle different noise values. Under overestimating conditions, on the other hand, the MPNN is able to achieve superior performances compared to SPADA, reaching an accuracy of 99% against 97%, respectively.

7) *MPNN vs SPADA - Performances on Different Noise Statistics*: This experiment has the aim of comparing the peak or absolute performances of MPNN and SPADA in case we have a training dataset with same statistics of the validation dataset. Understanding the maximum performances is fundamental to have an upper-bound on a real deployment and to know the learning capabilities of the algorithm/model.

In Fig. 11 we report the validation accuracy reached by MPNN and SPADA varying the adopted dataset and for different standard deviations of the noise. For the MPNN we use training and validation datasets with the same value of  $\sigma$ , while the standard deviation of the likelihood in the SPADA is the same as in the validation dataset. We notice that the absolute performances of MPNN outperform the classic SPADA for both  $\sigma = 0.1$  m and  $\sigma = 0.5$  m and for all datasets. Therefore, the proposed method is able to fully solve the problem and learn synthetic or realistic noise representations. Clearly, for the dataset obtained with PointPillars, we cannot tune the quantity of noise introduced by the ML model and consequently the red and blue circles for the dataset PointPillars coincide. Lastly, we can observe that the degradation of performances passing from  $\sigma = 0.1$  m to  $\sigma = 0.5$  m are worst for the MPNN. This behaviour is further investigated in the next experiment.

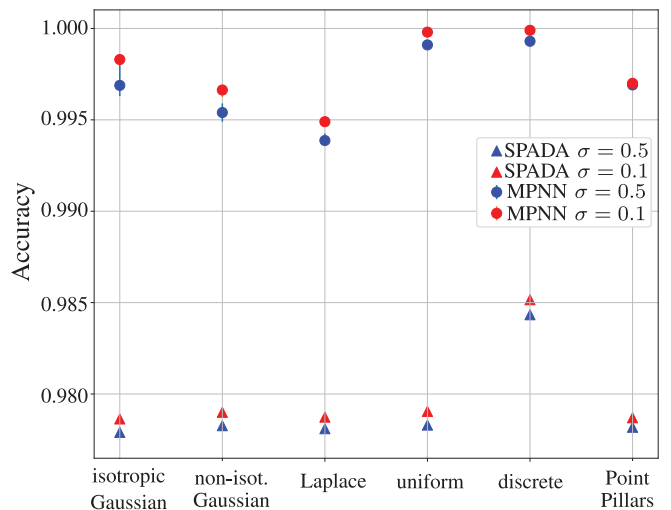


Fig. 11. Comparison of reached accuracy using MPNN (circles) and SPADA (triangles) in different validation datasets. Red markers represent an artificial measurement noise in the validation dataset with standard deviation 0.1 m, while the blue markers describe a standard deviation of 0.5 m. For the MPNN, the noise statistics of the training match those of the validation, whereas for SPADA the standard deviation of the likelihood matches the standard deviation of the noise in the validation dataset.

8) *MPNN vs SPADA - Performances on Different Noise Intensity*: In this last assessment, we study how the MPNN and SPADA perform over different levels of detection accuracies. This is useful to understand if there are conditions in which one method outperforms the other.

To this aim, we consider different standard deviations of the Gaussian measurement error. The results of this analysis are in Fig. 12, where we report the validation accuracy of the MPNN and SPADA in a scenario with 100 (Fig. 12(a)) or 50 (Fig. 12(b)) pedestrians to be detected. First, we observe that the performances for the scenario in Fig. 12(b) are generally higher than for the scenario in Fig. 12(a). This is due to the fact that with a higher number of pedestrians, the uncertainty on the data association increases and the data association becomes

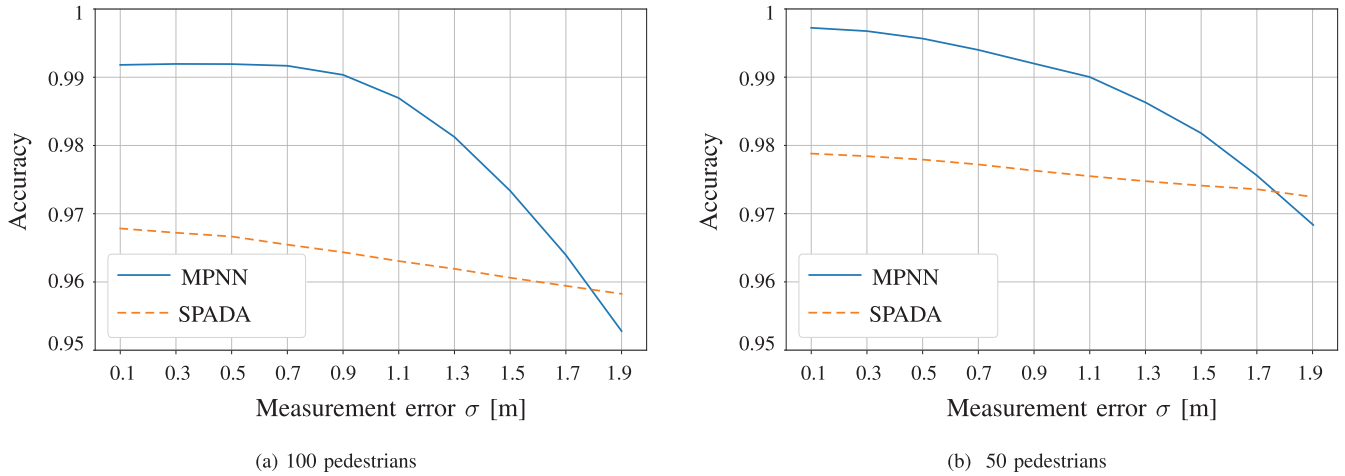


Fig. 12. Comparison of the impact of measurement error in terms of accuracy between MPNN and SPADA for a scenario with (a) 100 pedestrians or (b) 50 pedestrians. An isotropic Gaussian distribution with standard deviation  $\sigma$  is considered for the additive noise statistics.

more challenging. Second, comparing the two methods across different noise intensities, we note that the MPNN is preferable when the standard deviation of the noise is below 1.8 m in both scenarios. We believe that this behaviour is caused by the fact that the MPNN has difficulties in learning a noise with high variance with respect to a low power noise. On the contrary, the SPADA depends on the standard deviation of the likelihood that in this case is known a-priori and equal to the standard deviation of the validation dataset. Therefore, with high noise intensities, it is preferable to use SPADA as we would need too many samples to learn the noise directly from data.

## V. CONCLUSION

This paper addressed the problem of data association in a cooperative vehicular sensing scenario with multiple vehicles detecting pedestrians through lidar sensors. To solve the problem, we proposed an MPNN model based on a novel graph representation encoding node and edge feature attributes to express the detection knowledge. The validation was carried out in a vehicular environment simulated by CARLA software, which allows to reproduce realistic cooperative lidar sensing scenarios. We considered the PointPillars model for the extraction of bounding boxes from the lidar point cloud, obtaining realistic statistics of bounding boxes measurements. Furthermore, we compared the proposed method with the conventional SPADA to investigate the generalization capabilities and peak performances.

Results showed that the proposed MPNN model is able to learn the correct associations under several realistic measurement statistics and handles good generalization capabilities when it comes to dealing with untrained conditions, such as different measurement error statistics, noise intensities, number of vehicles and new scenarios. The lidar detection error introduced by PointPillars has been found to be well approximated by a Gaussian distribution with standard deviation equal to 10 cm. Under this condition, very high accuracy can be reached by training the model on artificial noises, e.g., Laplace or Gaussian, and then validate the model on the field

with realistic noise distribution produced by PointPillars. Concerning the comparison with the classic SPADA, we found that, under overestimation of noise intensity, the proposed method achieved higher performances. Moreover, regarding peak performances, MPNN completely outperforms SPADA up to a noise standard deviation of 1.8 m.

In the incoming years, the relevance of cooperative perception is expected to grow rapidly, particularly in the context of automated and connected mobility, where the new-generation V2X communication technologies bring opportunities for the development of new services. It follows that an efficient management of data association is a fundamental and crucial step for enabling cooperative sensing. As a result, we expect our work to be extended and applied to different contexts. By enhancing the data association performance, our method provides a solid foundation for more accurate and robust object tracking when combined with existing tracking algorithms which exploit the information shared by the vehicles to perform cooperative positioning or sensing of the surrounding environment.

A natural extension of the work would be to manage and account for possible false and/or missed detections through intra-temporal association and non fully-connected vehicular networks. Future developments could also embrace the area of distributed sensor networks in which the flood of information over sensors demands fast interactions of locally-available data but guarantees higher resilience compared to centralized architecture, overcoming the problem of single point of failure. On the other hand, hop-by-hop transport might introduce a non-negligible time delay before the same full information is available at all nodes. In addition, we plan to evaluate our method on real-world cooperative data which would help further validate and refine our approach, ensuring its effectiveness in addressing real-world object detection and tracking challenges.

## REFERENCES

- [1] A. Mahdavian, A. Shojaei, S. McCormick, T. Papandreou, N. Eluru, and A. A. Oloufa, "Drivers and barriers to implementation of connected, automated, shared, and electric vehicles: An agenda for future research," *IEEE Access*, vol. 9, pp. 22195–22213, Feb. 2021.

- [2] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. McCullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018.
- [3] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G evolution: A view on 5G cellular technology beyond 3GPP release 15," *IEEE Access*, vol. 7, pp. 127639–127651, Sep. 2019.
- [4] C. De Lima et al., "Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges," *IEEE Access*, vol. 9, pp. 26902–26925, Jan. 2021.
- [5] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133995–134030, Jul. 2020.
- [6] H. Bagheri et al., "5G NR-V2X: Toward connected and cooperative autonomous driving," *IEEE Commun. Standards Mag.*, vol. 5, no. 1, pp. 48–54, Mar. 2021.
- [7] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.
- [8] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Feb. 2012.
- [9] A. Ihler, J. Fisher, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.
- [10] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.
- [11] Y. Zheng, N. Cao, T. Wimalajeewa, and P. K. Varshney, "Compressive sensing based probabilistic sensor management for target tracking in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6049–6060, Nov. 2015.
- [12] N. Cao, S. Choi, E. Masazade, and P. K. Varshney, "Sensor selection for target tracking in wireless sensor networks with uncertainty," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5191–5204, Oct. 2016.
- [13] D. Caveney, "Cooperative vehicular safety applications," *IEEE Control Syst. Mag.*, vol. 30, no. 4, pp. 38–53, Aug. 2010.
- [14] G. Pocovi, M. Lauridsen, B. Soret, K. I. Pedersen, and P. Mogensen, "Automation for on-road vehicles: Use cases and requirements for radio design," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC)*, Sep. 2015, pp. 1–5.
- [15] A. Milan, S. H. Rezatofghi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 4225–4232.
- [16] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 2017, pp. 300–311.
- [17] S. Zhang et al., "Distributed direct localization suitable for dense networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1209–1227, Apr. 2020.
- [18] S. Safavi, U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed localization: A linear theory," *Proc. IEEE*, vol. 106, no. 7, pp. 1204–1223, Jul. 2018.
- [19] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for localization-of-things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov. 2019.
- [20] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. IEEE*, vol. 85, no. 1, pp. 6–23, Jan. 1997.
- [21] F. Meyer et al., "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.
- [22] J. Williams and R. Lau, "Approximate evaluation of marginal association probabilities with belief propagation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 2942–2959, Oct. 2014.
- [23] F. Meyer and M. Z. Win, "Scalable data association for extended object tracking," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 491–507, May 2020.
- [24] D. Gaglione, G. Soldi, P. Braca, G. De Magistris, F. Meyer, and F. Hlawatsch, "Classification-aided multitarget tracking using the sum-product algorithm," *IEEE Signal Process. Lett.*, vol. 27, pp. 1710–1714, Sep. 2020.
- [25] F. Meyer and M. Z. Win, "Data association for tracking extended targets," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 337–342.
- [26] F. Meyer, Z. Liu, and M. Z. Win, "Scalable probabilistic data association with extended objects," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.
- [27] M. Brambilla et al., "Cooperative localization and multitarget tracking in agent networks with the sum-product algorithm," *IEEE Open J. Signal Process.*, vol. 3, pp. 169–195, Mar. 2022.
- [28] H.-A. Loeliger, J. Dauwels, J. Hu, S. Kori, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, Jul. 2007.
- [29] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "A scalable algorithm for tracking an unknown number of targets using multiple sensors," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3478–3493, Mar. 2017.
- [30] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elect. Eng. F Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [31] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. OE8, no. 3, pp. 173–184, Jul. 1983.
- [32] R. W. Sittler, "An optimal data association problem in surveillance theory," *IEEE Trans. Mil. Electron.*, vol. MIL-8, no. 2, pp. 125–139, Apr. 1964.
- [33] R. Singer, R. Sea, and K. Housewright, "Derivation and evaluation of improved tracking filter for use in dense multitarget environments," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 4, pp. 423–432, Jul. 1974.
- [34] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Trans. Autom. Control*, vol. AC-23, no. 4, pp. 618–626, Aug. 1978.
- [35] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1388–1400, Apr. 2015.
- [36] M. Brambilla et al., "Sensors localization and target tracking in underwater environment via belief propagation," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Dec. 2021, pp. 641–646.
- [37] M. Brambilla, M. Nicoli, G. Soatti, and F. DeFlorio, "Augmenting vehicle localization by cooperative sensing of the driving environment: Insight on data association in urban traffic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1646–1663, Apr. 2020.
- [38] P. Sharma, A.-A. Saucan, D. J. Bucci, and P. K. Varshney, "Decentralized Gaussian filters for cooperative self-localization and multi-target tracking," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5896–5911, Nov. 2019.
- [39] S. Stankovic, N. Ilic, and M. S. Stankovic, "Adaptive consensus-based distributed system for multisensor multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 3, pp. 2164–2179, Dec. 2022.
- [40] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Mar. 2007.
- [41] G. Papa, R. Repp, F. Meyer, P. Braca, and F. Hlawatsch, "Distributed Bernoulli filtering using likelihood consensus," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 218–233, Jun. 2019.
- [42] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 1, pp. 57–71, Dec. 2016.
- [43] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of Driver Assistance Systems*. Amsterdam, The Netherlands: Springer International Publishing, 2014.
- [44] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila, and A. Rakotonirainy, "Perception, information processing and modeling: Critical stages for autonomous driving applications," *Annu. Rev. Contr.*, vol. 44, pp. 323–341, Oct. 2017.
- [45] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [46] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proc. 19th IEEE Conf. Decis. Contr. Symp. Adaptive Processes*, Dec. 1980, pp. 807–812.
- [47] D. Musicki and R. Evans, "Linear joint integrated probabilistic data association - LJIPDA," in *Proc. 41st IEEE Conf. Decis. Contr.*, Dec. 2002, vol. 3, pp. 2415–2420.
- [48] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.
- [49] R. Mahler, "'Statistics 102' for multisource-multitarget detection and tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 376–389, Jun. 2013.
- [50] M. Kalandros and L. Pao, "Multisensor covariance control strategies for reducing bias effects in interacting target scenarios," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 153–173, Jan. 2005.

- [51] D. Musicki and B. La Scala, "Multi-target tracking in clutter without measurement assignment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 877–896, Jul. 2008.
- [52] S. Nagappa and D. E. Clark, "On the ordering of the sensors in the iterated-corrector probability hypothesis density (PHD) filter," in *Proc. Signal Process., Sens. Fusion, Target Recognit. XX*, I. Kadar, Ed., Bellingham, WA, USA: SPIE, May 2011, vol. 8050, pp. 275–280.
- [53] R. Mahler, "Approximate multisensor CPHD and PHD filters," in *Proc. 13th Int. Conf. Inf. Fusion*, Jul. 2010, pp. 1–8.
- [54] M. Jiang, W. Yi, R. Hoseinnezhad, and L. Kong, "Distributed multi-sensor fusion using generalized multi-Bernoulli densities," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 1332–1339.
- [55] A.-A. Saucan, M. J. Coates, and M. Rabbat, "A multisensor multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5495–5509, Oct. 2017.
- [56] L. Gao, G. Battistelli, and L. Chisci, "Event-triggered distributed multitarget tracking," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 570–584, Sep. 2019.
- [57] M. Frohlc, C. Lindberg, K. Granstrom, and H. Wymeersch, "Multisensor poisson multi-Bernoulli filter for joint target-sensor state tracking," *IEEE Trans. Intell. Veh.*, vol. 4, no. 4, pp. 609–621, Dec. 2019.
- [58] S. He, H.-S. Shin, and A. Tsourdos, "Distributed joint probabilistic data association filter with hybrid fusion strategy," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 286–300, Jan. 2020.
- [59] A. K. Gostar et al., "Centralized cooperative sensor fusion for dynamic sensor network with limited field-of-view via labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 69, pp. 878–891, Dec. 2021.
- [60] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Jul. 2006, vol. 1, pp. 666–673.
- [61] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vision.*, Marseille, France: Springer, Oct. 2008, pp. 788–801.
- [62] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Aug. 2008, pp. 1–8.
- [63] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using  $k$ -shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Feb. 2011.
- [64] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *Int. J. Comput. Vision.*, vol. 75, no. 2, pp. 247–266, Aug. 2007.
- [65] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jul. 2012, pp. 1815–1821.
- [66] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 81–102, Jan. 2009.
- [67] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Dec. 2009.
- [68] V. Garcia Satorras and M. Welling, "Neural enhanced belief propagation on factor graphs," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, A. Banerjee and K. Fukumizu, Eds., San Diego, CA, USA: PMLR, Apr. 2021, vol. 130, pp. 685–693.
- [69] M. Liang and F. Meyer, "Neural enhanced belief propagation for cooperative localization," in *Proc. IEEE Statist. Signal Process. Workshop (SSP)*, Aug. 2021, pp. 326–330.
- [70] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Sep. 2020.
- [71] K. Yoon et al., "Inference in probabilistic graphical models by graph neural networks," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.* Pacific Grove, CA, USA: IEEE, Nov. 2019, pp. 868–875.
- [72] G. Braso and L. Leal-Taixe, "Learning a neural solver for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, Jun. 2020, pp. 6246–6256.
- [73] L. Barbieri, B. C. Tedeschini, M. Brambilla, and M. Nicoli, "Implicit vehicle positioning with cooperative lidar sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.
- [74] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vision., Tel Aviv, Israel: Springer*, 2022, Oct. pp. 1–21.
- [75] B. C. Tedeschini, M. Brambilla, L. Barbieri, and M. Nicoli, "Addressing data association by message passing over graph neural networks," in *Proc. 25th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2022, pp. 1–7.
- [76] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, Mountain View, CA, USA: PMLR, Nov. 2017, pp. 1–16.
- [77] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.
- [78] H. Su et al., "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2530–2539.
- [79] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11037–11045.
- [80] M. Liang and F. Meyer, "Neural enhanced belief propagation for multiobject tracking," Dec. 2022, *arXiv:2212.08340*.
- [81] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.* Sydney, Australia: PMLR, Aug. 2017, pp. 1263–1272.
- [82] P. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," Oct. 2018, *arXiv:1806.01261*.
- [83] P. Battaglia et al., "Interaction networks for learning about objects, relations and physics," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS Conf.)*, Oct. 2016, vol. 29, pp. 4502–4510.
- [84] D. K. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS Conf.)*, Dec. 2015, vol. 28, pp. 2224–2232.
- [85] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS Conf.)*, Oct. 2016, vol. 29, pp. 2244–2252.
- [86] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS Conf.)*, Dec. 2017, vol. 30, pp. 5998–6008.
- [87] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," Feb. 2017, *arXiv:1707.01926*.
- [88] Z. Cui, K. Henriksson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.
- [89] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.
- [90] E. Arnold, S. Mozaffari, and M. Dianati, "Fast and robust registration of partially overlapping point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1502–1509, Apr. 2022.
- [91] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017, *arXiv:1412.6980*.
- [92] S. Luo and W. Hu, "Score-based point cloud denoising," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, Oct. 2021, pp. 4583–4592.



**Bernardo Camajori Tedeschini** (Graduate Student Member, IEEE) received the B.Sc. (Hons.) in computer science and M.Sc. (Hons.) degrees in telecommunications engineering from the Politecnico di Milano, Italy, in 2019 and 2021, respectively. From November 2021, he started as a Ph.D. fellow in information technology with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano. He is currently a Visiting Researcher with the Laboratory for Information & Decision Systems at the Massachusetts Institute of

Technology (MIT), Cambridge, MA, USA. His research interests include federated learning, machine learning, and localization methods. He was a recipient of the Ph.D. grant from the ministry of the Italian government Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) and the Roberto Rocca Doctoral Fellowship granted by MIT and Politecnico di Milano.



**Mattia Brambilla** (Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunication engineering, and the Ph.D. degree (*cum laude*) in information technology from the Politecnico di Milano, in 2015, 2017, and 2021, respectively. He was a Visiting Researcher with the NATO Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, in 2019. In 2021, he joined as Research Fellow with the Faculty of Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB) at the Politecnico di Milano. His research interests include signal processing, statistical learning, and data fusion for cooperative localization and communication. He was the recipient of the Best Student Paper Award at the 2018 IEEE Statistical Signal Processing Workshop.



**Luca Barbieri** (Member, IEEE) received the B.Sc. and M.Sc. (*cum laude*) degrees in telecommunication engineering and the Ph.D. degree (*cum laude*) in information technology from the Politecnico di Milano, in 2017, 2019, and 2023, respectively. He was a Visiting Researcher with the King's Communications, Learning & Information Processing (KCLIP) laboratory at King's College London, London, U.K., in 2022. He is currently a Research Assistant with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano. His current research interests focus on machine learning, federated learning, and localization methods for vehicular and industrial networks.



**Gabriele Balducci** received the M.Sc. degree (*cum laude*) in telecommunication engineering with specialization in Signal and Data Analysis, from Politecnico di Milano, in 2022. In the same year, he concluded his path as an Alta Scuola Politecnica student, XVII cycle. Currently, he is an employee in R&D at Nokia Italia cultivating his passion for Statistical Signal Processing.



**Monica Nicoli** (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in communication engineering from the Politecnico di Milano, Milan, Italy, in 1998 and 2002, respectively. She was a Visiting Researcher with ENI Agip, from 1998 to 1999, and Uppsala University, in 2001. In 2002, she joined as a Faculty Member with the Politecnico di Milano. She is currently an Associate Professor in telecommunications with the Department of Management, Economics and Industrial Engineering. Her research interests include signal processing, machine learning, and wireless communications, with emphasis on smart mobility and Internet of Things (IoT) applications. She was a recipient of the Marisa Bellisario Award, in 1999, and a corecipient of the best paper awards of the IEEE Symposium on Joint Communications and Sensing, in 2021, the IEEE Statistical Signal Processing Workshop, in 2018, and the *IET Intelligent Transport Systems* journal, in 2014. She is an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. She has also served as an Associate Editor for the *EURASIP Journal on Wireless Communications and Networking*, from 2010 to 2017, and a Lead Guest Editor for the Special Issue on Localization in Mobile Wireless and Sensor Networks, in 2011.

Open Access funding provided by 'Politecnico di Milano' within the CRUI CARE Agreement.