Multi-Agent Reinforcement Learning for Distributed Cooperative Vehicular Positioning

Bernardo Camajori Tedeschini, *Graduate Student Member, IEEE*, Mattia Brambilla, *Member, IEEE*, Monica Nicoli, *Senior Member, IEEE*, and Moe Z. Win, *Fellow, IEEE*

***Abstract*—With the advent of cooperative intelligent transport systems (C-ITS) and vehicle-to-everything (V2X) communications, cooperative positioning based on V2X sharing of location information has been emerging as a promising augmentation system for conventional satellite navigation. An example is implicit cooperative positioning (ICP) which relies on Bayesian filtering for cooperative sensing of targets that are used as reference points for improving vehicle positioning. ICP methods, however, rely on predetermined models which makes them sub-optimal in case of non-Gaussian non-linear models or complex cooperation graphs. To address these limitations, the paper proposes a decentralized-partially observable Markov decision process (Dec-POMDP) framework, paired with deep multi-agent reinforcement learning (MARL) algorithms. We introduce a novel ICP-multi-agent proximal policy optimization (MAPPO) algorithm where distributed agents (i.e., vehicles) dynamically activate/deactivate the radio links for cooperation with the neighbors to optimize the communication efficiency, still guaranteeing accurate positioning. We reproduce a realistic C-ITS scenario with CARLA simulator, where vehicles move according to real-world dynamics and communicate with each other to cooperatively sense their locations. Results show that the proposed ICP-MAPPO algorithm, with its dynamic-decentralized-execution and centralized-training schemes, outperforms state-of-the-art ICP methods by 21% in terms of positioning accuracy, and it can reduce the communication overhead by following the optimal learned policy.**

***Index Terms*—MARL, Dec-POMDP, implicit cooperative positioning, Bayesian-filtering, message passing algorithm.**

Received 21 June 2024; revised 7 September 2024; accepted 19 September 2024. Date of publication 30 September 2024; date of current version 17 October 2025. The fundamental research described in this paper was supported, in part, by the Roberto Rocca Doctoral Fellowship granted by the Massachusetts Institute of Technology and Politecnico di Milano, by MOST – Sustainable Mobility National Research Center European Union Next-GenerationEU (piano nazionale di ripresa e resilienza (PNRR) – missione 4 componente 2, investimento 1.4 – D.D. 1033 17/06/2022, CN00000023), by the National Science Foundation under Grant CNS-2148251, and by the federal agency and industry partners in the RINGS Program. An earlier version of this paper was presented in part at the International Conference on Information Fusion (FUSION), Venice, Italy, July 2024 [DOI: 10.23919/FUSION59988.2024.10706524]. *(Corresponding author: Moe Z. Win.)*

Bernardo Camajori Tedeschini is with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy, and also with the Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: bernardo.camajori@polimi.it).

Mattia Brambilla is with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy (e-mail: mattia.brambilla@polimi.it).

Monica Nicoli is with the Dipartimento di Ingegneria Gestionale (DIG), Politecnico di Milano, 20156 Milan, Italy (e-mail: monica.nicoli@polimi.it).

Moe Z. Win is with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: moewin@mit.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3471909.

Digital Object Identifier 10.1109/TIV.2024.3471909

I. Introduction

Cooperative positioning (CP) represents a key enabling feature for future automated mobility services [1], [2], [3], [4], [5], [6], [7], [8]. Automated vehicles leverage an onboard sensor suite including global navigation satellite systems (GNSS), light detection and ranging (LIDAR), radio detection and ranging (RADAR), and stereo cameras to perceive the surrounding environment and perform automated maneuvers [9], [10], [11], [12], [13]. At today, these sensors are not yet able to guarantee high-precision localization in harsh environments such as dense urban areas or canyons and this is a main issue for autonomous driving functions [14]. Recently, methods have been proposed to combine localization sensors with the latest 5th generation (5G) of cellular communications [15], [16], [17], [18], [19], [20], depicting a new horizon for mobile connectivity and positioning services [21], [22], [23], [24]. 5G vehicle-to-everything (V2X) communications are envisioned as crucial in the evolution towards cooperative intelligent transport systems (C-ITS) [25], [26], [27], [28] by enabling simultaneous communication and localization functionalities [29], [30], [31]. CP among vehicles, by means of sidelink V2X communications, can be used to overcome the GNSS performance degradation and guarantee a seamless high-accuracy positioning (HAP) service [32], [33], [34], [35], [36]. The complexity lies in the resource-intensive nature of CP [37], which involves vehicles interacting with each other repeatedly to determine positions. In particular, this cooperative process demands significant power and bandwidth [38], [39], [40], while also facing challenges in scheduling transmissions due to the intricate measurement and information fusion processes [41], [42], [43]. These factors may cause larger delays and scalability issues in cooperative localization [44], [45].

An emerging approach for cooperative vehicle localization is implicit cooperative positioning (ICP) [32], [46], which integrates GNSS and onboard passive sensor data through Bayesian-filtering, e.g., conventional extended Kalman filter (EKF) or message passing algorithm (MPA), to coherently fuse the measurements at different vehicles. In ICP, passive objects such as poles, road signs, or traffic lights, are cooperatively detected by multiple vehicles and exploited as noisy anchor points to enhance the vehicle location accuracy. In case of a centralized data-processing architecture gathering all vehicles' measurements, convergence can be achieved, but at the expense of high computational complexity. Standard MPA algorithms enable decentralized processing but are optimal only in case of Gaussian-linear models and acyclic factor graphs [47], [48], [49], [50]. Recent solutions tried to limit the aforementioned problems by either performing fully-distributed particle-based MPA between vehicles [34] or auto-adjusting the parameters of time-varying models [51]. Still, they rely on particle-based solutions which require high communication and computational loads which limit their scalability.

In recent years, there has been a growing reliance on machine learning (ML) tools to overcome the limits of conventional approaches, especially regarding scalability and non-linear models [52], [53], [54], [55]. In particular, the reinforcement learning (RL) paradigm [56], [57], [58] and its deep learning (DL)-based version [59], [60], [61] are notably effective in challenging single-agent Markov decision processs (MDPs) where labeled data are scarce or costly. They also excel in environments where the agent's actions directly impact the state of the environment and long-term rewards are prioritized [62], [63], [64]. Indeed, RL can be seen as a generalization of Bayesian filtering where the agents do not just predict the state through belief computation but also make decisions to optimize the cooperative process by maximizing long-term rewards, with a policy guiding the decision from state to action. RL is especially well-suited for complex scenarios with extensive state and action spaces, where deep neural networks (DNNs) can efficiently approximate the high-dimensional, nonlinear functions that represent such policies [59], [65]. This approach has been successfully applied in several fields, varying from rate and power control [66], [67], [68], [69] to dynamic spectrum access in multi-user scenarios and efficient scheduling in vehicular networks [70], [71], [72], [73].

In case more than one agent acts in the environment and the state is not directly observable, we categorize the framework as multi-agent RL (MARL) [74] and the system as decentralized-partially observable MDP (Dec-POMDP) [75], [76], [77]. MARL involves independent agents whose actions influence each other's perception of the environment, and it is often solved with the usage of recurrent neural network (RNN), exploiting histories of observations and actions [78]. MARL algorithms, similarly to RL methods, can be divided into two categories: Q-learning and policy optimization (PO) (which comprises actor-critic methods) [79], [80], [81]. Q-learning focuses on estimating the long-term reward (i.e., Q-value) of each action, selecting the action with the highest Q-value and indirectly (i.e., not explicitly) formulating the policy [82], [83], [84]. On the other hand, PO directly optimizes the policy through the gradient of the total reward relative to policy parameters [85], [86], [87], [88]. Multi-agent PO algorithms, especially when combined with a centralized agent learning and a decentralized execution of the policies (e.g., multi-agent proximal policy optimization (MAPPO) [85]), have shown remarkable performances with respect to Q-learning algorithms. This is mainly due to their being free of learning biases and improved sampling efficiency thanks to training guidelines like parameter sharing [89], [90], [91].

First attempts to employ MARL for CP focus on target tracking by intelligent and connected unmanned aerial vehicles (UAVs) [92] or on agent scheduling for improving CP [93]. In [92], the RL objective was to maneuver the agents to track passive objects. However, they considered the state (i.e., the location) of the agents as perfectly known, while the main challenge was to estimate from the measurements their state jointly with target sensing. In [93], the agent state was estimated with conventional MPA, while the RL objective was to activate/deactivate links between agents to optimize cooperative positioning performances (i.e., by minimizing the positioning error bound (PEB)). The drawbacks of this method are that RL is not actively used for positioning but rather as an assistance method to MPA, and that they consider one agent only, i.e., a single link, at the time instead of exploiting the full potential of multi-agent systems (MASs).

Overall, the fundamental unresolved questions related to CP are as follows: i) how to design a decentralized MARL algorithm that simultaneously performs the computation of the agent state beliefs and the scheduling of the agent-to-agent communication resources, optimizing both location accuracy and communication efficiency; ii) what positioning accuracy improvement can be achieved with respect to state-of-the-art Bayesian approaches like ICP that exploit passive object detections between multiple agents; iii) what are the main trade-offs between positioning improvement and communication resource optimization. Addressing these questions is mandatory for the employment in connected automated vehicles (CAVs), in particular to ensure scalability and handle real-word impairments encountered in vehicular scenarios. In this perspective, the goals of this paper are to develop agent-specific policies for communication scheduling between neighbors and, at the same time, learning a representation of the system dynamics that takes advantage of the selected neighbors' measurements. We propose a MARL-based ICP, a new paradigm in which PO RL algorithms are exploited to extend the conventional Bayesian-filtering approach incorporating the actions of the agents. The main idea is to learn from data the relation between agents' states and passive feature observations (see Fig. 1 for a representation of the cooperative scenario) by selecting for the cooperation only those links to the neighbors that can provide a significant gain to the positioning accuracy. This approach is shown to not only improve the localization performance but also enhance the communication efficiency. In this paper, we propose a new MARL algorithm, namely ICP-MAPPO, expressly designed for performing efficient distributed positioning through the MARL framework and extending the conventional Bayesian-filtering ICP to data-driven approaches. The key contributions are as follows:

- We revise the ICP Bayesian-filtering approach analyzing the current limitations and investigating more general frameworks for solution, drawing from the Dec-POMDP system model and MARL methods.
- We reformulate the ICP methodology into a MARL problem and we design the new ICP-MAPPO solution, relying on dynamic-decentralized-execution and training schemes to simultaneously optimize the Bayesian-filtering and MARL objectives.
- We validate the proposed ICP-MAPPO approach in a realistic C-ITS scenario simulated with CARLA [94], where CAVs perform CP by cooperatively localizing

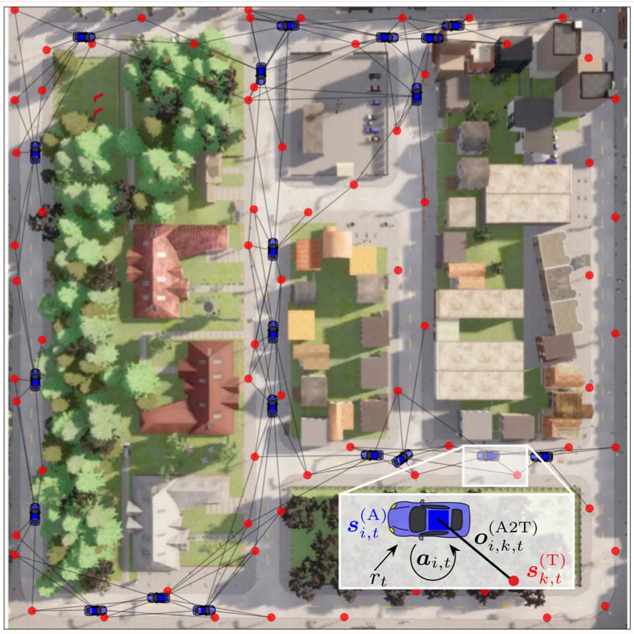


Fig. 1. Cooperative positioning scenario with twenty vehicles (blue vehicle icons), sensed poles acting as ancors (red circles) and detections (black lines).

TABLE I
MAIN ABBREVIATIONS

Acronym	Definition
A2A	Agent-to-agent
A2T	Agent-to-target
Dec-POMDP	Decentralized-partially observable Markov decision process
EKF	Extended Kalman Filter
ICP	Implicit cooperative positioning
LSTM	Long short-term memory
MLP	Multi-layer perceptron
MAPPO	Multi-agent proximal policy optimization
MARL	Multi-agent reinforcement learning
MPA	Message passing algorithm

passive targets, i.e., poles, distributed over the scene and acting as anchors.

- We perform a comparison with the state-of-the-art ICP algorithm [32] and single-agent-based algorithms. We prove the superior performances of the proposed algorithm both in terms of positioning error and communication efficiency.

For easy reference, Table I lists the main abbreviations used throughout the paper. The rest of this paper is structured as follows. Section II describes the system model of cooperative agents. Section III reviews the ICP Bayesian-filtering. Section IV presents the MARL framework and the proposed ICP-MAPPO execution and training schemes. Section V provides information about the simulated scenario and the results. Finally, Section VI draws the conclusions.

TABLE II
LIST OF NOTATIONS

Notation	Definition
N, K	Number of agents and passive objects
$\mathbf{s}_{i,t}$, $\mathbf{a}_{i,t}$, $\mathbf{o}_{i,t}$	State, action and observation of agent i at time t
$\mathbf{h}^{\mathrm{b}}_{i,t}$, $\mathbf{h}^{\mathrm{V}}_{i,t}$	History in belief and critic NNs of agent i at time t
$\boldsymbol{\tau}$, $\boldsymbol{\tau}_t$	Trajectory and transition at time t
r_t, R_t	Reward and reward-to-go at time t
$\pi_{\boldsymbol{\theta}}$, $V_{\boldsymbol{\phi}}$, $b_{\boldsymbol{\psi}}$	Actor, critic and beliefs NNs
H, L_τ	Horizon and trajectory length
$A_{i,t}$	Advantage function of agent i at time t
α, β, ϵ	Entropy, reward and clipping coefficients
γ, μ	Discount factor and learning rate

I. Notations

Random variables are displayed in sans serif, upright fonts; their realizations in serif, italic fonts. Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. For example, a random variable and its realization are denoted by x and x; a random vector and its realization are denoted by $\mathbf{x}$ and $\boldsymbol{x}$; a random matrix and its realization are denoted by $\mathbf{X}$ and $\boldsymbol{X}$, respectively. Random sets and their realizations are denoted by up-right sans serif and calligraphic font, respectively. For example, a random set and its realization are denoted by X and $\mathcal{X}$, respectively. The function $p_{\mathsf{x}}(x)$, and simply $p(x)$ when there is no ambiguity, denotes the probability density function (PDF) of x. Notations $\boldsymbol{X}^\top$, $\boldsymbol{X}^*$ and $\boldsymbol{X}^{\mathrm{H}}$ indicate the matrix transposition, conjugation and conjugate transposition. With the notation $\mathsf{x} \sim \mathcal{N}(\mu, \sigma^2)$ we indicate a Gaussian random variable x with mean μ and standard deviation σ, whose PDF is denoted by $\mathcal{N}(x; \mu, \sigma^2)$. We use $\mathbb{E}\{\cdot\}$ and $\mathbb{V}\{\cdot\}$ to denote the expectation and the variance of a random variable, respectively. $\mathbb{R}$ and $\mathbb{C}$ stand for the set of real and complex numbers, respectively. Finally, we define with blockdiag$(\cdot)$ the block diagonal matrix whose diagonal contains the input blocks matrices.

Notations and definitions of important quantities used in the paper are summarized in Table II.

II. System Model

We consider a vehicular network where a set of N vehicles engage in cooperative localization as depicted in Fig. 1. The connectivity graph for vehicle cooperation at time t is $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, with $\mathcal{V} = \{1, 2, \ldots, N\}$ representing the set of agents (vehicles), and $\mathcal{E}_t$ the edges (communication links) among them. Each agent $i \in \mathcal{V}$ in the network at time t has a set of neighbors $\mathcal{N}_{i,t}$, and it is assigned a state $\mathbf{s}^{(\mathrm{A})}_{i,t} = [\mathbf{u}^{(\mathrm{A})^\top}_{i,t}\ \mathbf{v}^{(\mathrm{A})^\top}_{i,t}]^\top$, where $\mathbf{u}^{(\mathrm{A})}_{i,t}$ and $\mathbf{v}^{(\mathrm{A})}_{i,t}$ are the 2D position and velocity vectors, respectively, defined in a global coordinate system. We denote with $\mathbf{s}^{(\mathrm{A})}_t = [\mathbf{s}^{(\mathrm{A})}_{i,t}]_{i=1}^N$ the aggregate state of all the vehicles at time t. The kinematic state transition of vehicle i at time t is modelled as

$$\mathbf{s}^{(\mathrm{A})}_{i,t} = f^{(\mathrm{A})}\left(\mathbf{s}^{(\mathrm{A})}_{i,t-1}, \mathbf{w}^{(\mathrm{A})}_{i,t-1}\right) \quad (1)$$

where $f^{(\mathrm{A})}(\cdot)$ is is a nonlinear function that governs the dynamics of the vehicle's state and $\mathbf{w}_{i,t-1}^{(\mathrm{A})}$ represents the driving noise process, incorporating the uncertainty in motion. The model in (1) is associated to a state-transition PDF denoted as $T\big(\boldsymbol{s}_{i,t}^{(\mathrm{A})}|\boldsymbol{s}_{i,t-1}^{(\mathrm{A})}\big) \triangleq p\big(\boldsymbol{s}_{i,t}^{(\mathrm{A})}|\boldsymbol{s}_{i,t-1}^{(\mathrm{A})}\big)$.

The scenario includes a set $\mathcal{F} = \{1, 2, \ldots, K\}$ of K static and passive objects (or targets, denoted as red circles in Fig. 1) that vehicles can detect and localize by on-board sensors. To facilitate detection by vehicle sensors, specific objects easily identifiable and suitable for the purpose should be used. In this study, poles have been selected due to their ubiquity (especially in urban areas), ease of recognition, and fixed nature. Each pole k is described by a 2D position state $\mathbf{s}_{k,t}^{(\mathrm{T})}$, which is assumed to be constant over time. As before, we denote with $\mathbf{s}_t^{(\mathrm{T})} = [\mathbf{s}_{k,t}^{(\mathrm{T})}]_{k\in\mathcal{F}}$ the aggregate state of all passive objects at time t.

Vehicles are equipped with three distinct types of sensors. The first is a GNSS receiver, providing an estimate of the vehicle's state $\mathbf{s}_{i,t}^{(\mathrm{A})}$, modelled as

$$\mathbf{o}_{i,t}^{(\mathrm{GNSS})} = \boldsymbol{H}^{(\mathrm{GNSS})}\,\mathbf{s}_{i,t}^{(\mathrm{A})} + \mathbf{n}_{i,t}^{(\mathrm{GNSS})} \tag{2}$$

where $\mathbf{n}_{i,t}^{(\mathrm{GNSS})} \sim \mathcal{N}(\mathbf{0}_{2\times 2}, \boldsymbol{R}_{i,t}^{(\mathrm{GNSS})}) \in \mathbb{R}^{2\times 1}$ is a zero-mean Gaussian noise with covariance $\boldsymbol{R}_{i,t}^{(\mathrm{GNSS})} = {\sigma^{(\mathrm{GNSS})}}^2\boldsymbol{I}_2$, and $\boldsymbol{H}^{(\mathrm{GNSS})} = [\boldsymbol{I}_2\,\mathbf{0}_{2\times 2}] \in \mathbb{R}^{2\times 4}$. From (2), we define the GNSS likelihood function as $p(\boldsymbol{o}_{i,t}^{(\mathrm{GNSS})}|\boldsymbol{s}_{i,t}^{(\mathrm{A})})$, and with $\mathbf{o}_t^{(\mathrm{GNSS})} = [\mathbf{o}_{i,t}^{(\mathrm{GNSS})}]_{i=1}^N$ the aggregate GNSS measurements of all the vehicles at time t.

The second sensor refers to an active sensing technology for sidelink positioning offering relative agent-to-agent (A2A) location measurements for any pair of vehicles $(i, j) \in \mathcal{E}_t$

$$\mathbf{o}_{i,j,t}^{(\mathrm{A2A})} = \boldsymbol{H}^{(\mathrm{A2A})}\big(\mathbf{s}_{i,t}^{(\mathrm{A})} - \mathbf{s}_{j,t}^{(\mathrm{A})}\big) + \mathbf{n}_{i,j,t}^{(\mathrm{A2A})} \tag{3}$$

where $\boldsymbol{H}^{(\mathrm{A2A})} = [\boldsymbol{I}_2\,\mathbf{0}_{2\times 2}] \in \mathbb{R}^{2\times 4}$ and $\mathbf{n}_{i,j,t}^{(\mathrm{A2A})} \sim \mathcal{N}(\mathbf{0}_{2\times 2}, \boldsymbol{R}_{i,j,t}^{(\mathrm{A2A})})$ is a zero-mean Gaussian noise with covariance $\boldsymbol{R}_{i,j,t}^{(\mathrm{A2A})} = {\sigma^{(\mathrm{A2A})}}^2\boldsymbol{I}_2$. Additionally, agents have the capability to communicate with their neighbors to share location-related data.

The third sensor type is a passive technology (e.g., RADAR, LIDAR, camera, or any combination), used by vehicle i to detect a set of passive objects $\mathcal{F}_{i,t} \subseteq \mathcal{F}$ in proximity at time t, and produce agent-to-target (A2T) measurements for each object $k \in \mathcal{F}_{i,t}$ as

$$\mathbf{o}_{i,k,t}^{(\mathrm{A2T})} = \boldsymbol{H}^{(\mathrm{A2T})}\mathbf{s}_{i,t}^{(\mathrm{A})} - \mathbf{s}_{k,t}^{(\mathrm{T})} + \mathbf{n}_{i,k,t}^{(\mathrm{A2T})} \tag{4}$$

where $\boldsymbol{H}^{(\mathrm{A2T})} = [\boldsymbol{I}_2\,\mathbf{0}_{2\times 2}] \in \mathbb{R}^{2\times 4}$ and $\mathbf{n}_{i,k,t}^{(\mathrm{A2T})} \sim \mathcal{N}(\mathbf{0}_{2\times 2}, \boldsymbol{R}_{i,k,t}^{(\mathrm{A2T})})$ is a zero-mean Gaussian noise with covariance $\boldsymbol{R}_{i,k,t}^{(\mathrm{A2T})} = {\sigma^{(\mathrm{A2T})}}^2\boldsymbol{I}_2$.

We denote with $p(\boldsymbol{o}_{i,j,t}^{(\mathrm{A2A})}|\boldsymbol{s}_{i,t}^{(\mathrm{A})}, \boldsymbol{s}_{j,t}^{(\mathrm{A})})$ and $p(\boldsymbol{o}_{i,k,t}^{(\mathrm{A2T})}|\boldsymbol{s}_{i,t}^{(\mathrm{A})}, \boldsymbol{s}_{k,t}^{(\mathrm{T})})$ the A2A and A2T likelihoods, respectively. Moreover, we denote with $\mathbf{o}_{i,t} = [{\mathbf{o}_{i,t}^{(\mathrm{GNSS})}}^\top {\mathbf{o}_{i,t}^{(\mathrm{A2A})}}^\top {\mathbf{o}_{i,t}^{(\mathrm{A2T})}}^\top]^\top$ the vector of all available measurements of vehicle i at time t, where $\mathbf{o}_{i,t}^{(\mathrm{A2A})} = [\mathbf{o}_{i,j,t}^{(\mathrm{A2A})}]_{j\in\mathcal{N}_{i,t}}$ and $\mathbf{o}_{i,t}^{(\mathrm{A2T})} = [\mathbf{o}_{i,k,t}^{(\mathrm{A2T})}]_{k\in\mathcal{F}_{i,t}}$. The total number of unique A2A and A2T measurements at time t is defined as $N_t^{(\mathrm{A2A})} = \sum_{i=1}^N |\mathcal{N}_{i,t}|$ and $N_t^{(\mathrm{A2T})} = \sum_{i=1}^N |\mathcal{F}_{i,t}|$, respectively. Note that the A2A measurements are not subject to measurement-origin uncertainty, i.e., it is not requested to perform any data association algorithm for pairing them, as the enabling technology is assumed to be active. On the other hand, the A2T observations are unlabelled, as it is unknown which object gives rise to a measurement, being them produced by a passive sensing technology (e.g., RADAR or LIDAR). In this work, we assume that data association has already been performed at the vehicles (using, e.g., methods [53]) and that each A2T measurement has been correctly labeled with the originating target. We consider perfect data association as we aim to derive the best-case performances on the achievable accuracy of data-driven ICP and compare it with conventional Bayesian ICP in the same conditions. Interested readers can refer to [46] for details on data association and their impact on inference algorithms.

III. Bayesian Filtering

In this section, we describe the Bayesian filtering solution, under the ICP framework, and then we highlight its main drawbacks and improvements.

A. Centralized Implicit Cooperative Positioning

The objective of ICP is to concurrently estimate the state of all vehicles and passive objects in the network. To this aim, we define the set of all available measurements at time t as

$$\mathbf{o}_t = \boldsymbol{H}\,\mathbf{s}_t + \mathbf{n}_t \tag{5}$$

where $\mathbf{o}_t = [\mathbf{o}_{i,t}]_{i\in\mathcal{V}} \in \mathbb{R}^{(2N+2N_t^{(\mathrm{A2A})}+2N_t^{(\mathrm{A2T})})\times 1}$, $\boldsymbol{H}$ is the matrix modeling the relation to the states, defined as in [32], and $\mathbf{s}_t = [{\mathbf{s}_t^{(\mathrm{A})}}^\top {\mathbf{s}_t^{(\mathrm{T})}}^\top]^\top \in \mathbb{R}^{(4N+2K)\times 1}$ is the aggregated state of the system. $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t)$ is the overall measurement noise with covariance

$$\boldsymbol{R}_t = \mathrm{blockdiag}\big(\boldsymbol{R}_t^{(\mathrm{GNSS})},\ \boldsymbol{R}_t^{(\mathrm{A2A})}, \boldsymbol{R}_t^{(\mathrm{A2T})}\big)$$

where

$$\boldsymbol{R}_t^{(\mathrm{GNSS})} = \mathrm{blockdiag}\big(\boldsymbol{R}_{1,t}^{(\mathrm{GNSS})},\ \boldsymbol{R}_{2,t}^{(\mathrm{GNSS})}, \ldots, \boldsymbol{R}_{N,t}^{(\mathrm{GNSS})}\big)$$
$$\boldsymbol{R}_t^{(\mathrm{A2A})} = \mathrm{blockdiag}\Big(\boldsymbol{R}_{1,t}^{(\mathrm{A2A})},\ \boldsymbol{R}_{2,t}^{(\mathrm{A2A})}, \ldots, \boldsymbol{R}_{N_t^{(\mathrm{A2A})},\,t}^{(\mathrm{A2A})}\Big)$$
$$\boldsymbol{R}_t^{(\mathrm{A2T})} = \mathrm{blockdiag}\Big(\boldsymbol{R}_{1,t}^{(\mathrm{A2T})},\ \boldsymbol{R}_{2,t}^{(\mathrm{A2T})}, \ldots, \boldsymbol{R}_{N_t^{(\mathrm{A2T})},\,t}^{(\mathrm{A2T})}\Big)$$

with the ℓ-th entries given by $\boldsymbol{R}_{\ell,t}^{(\mathrm{A2A})} = \boldsymbol{R}_{i_\ell,j_\ell,t}^{(\mathrm{A2A})}$ and $\boldsymbol{R}_{\ell,t}^{(\mathrm{A2T})} = \boldsymbol{R}_{i_\ell,k_\ell,t}^{(\mathrm{A2T})}$.

The overall state estimate $\widehat{\boldsymbol{s}}_t$ is obtained through the minimum mean square error (MMSE) estimator as

$$\widehat{\boldsymbol{s}}_t = \mathbb{E}\{\mathbf{s}_t|\mathbf{o}_{1:t}\} = \int \boldsymbol{s}_t\,p\left(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}\right)\,d\boldsymbol{s}_t \tag{6}$$

where $\boldsymbol{o}_{1:t} = [\boldsymbol{o}_{t'}]_{t'=1}^{t}$ is the set of all aggregated measurements up to time t and $p(\boldsymbol{s}_t|\boldsymbol{o}_{1:t})$ is the posterior PDF defined as [95]

$$p\left(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}\right) \propto p\left(\boldsymbol{o}_t|\boldsymbol{s}_t\right) \int p\left(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}\right) p\left(\boldsymbol{s}_{t-1}|\boldsymbol{o}_{1:t-1}\right) d\boldsymbol{s}_{t-1}\,. \tag{7}$$

We denote with $b(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{1:t}) \triangleq p(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{1:t})$ the marginal posterior PDF, also called *belief* of agent i. Given that all the measurements are mutually independent, the likelihood function of $\boldsymbol{s}_t$ is computed as

$$\begin{aligned} p\left(\boldsymbol{o}_t|\boldsymbol{s}_t\right) = p\big(\boldsymbol{o}_t^{(\mathrm{GNSS})}|\boldsymbol{s}_t^{(\mathrm{A})}\big) \prod_{i=1}^{N} \prod_{j\in\mathcal{N}_{i,t}} p\big(\boldsymbol{o}_{i,j,t}^{(\mathrm{A2A})}|\boldsymbol{s}_{i,t}^{(\mathrm{A})}, \boldsymbol{s}_{j,t}^{(\mathrm{A})}\big) \\ \times \prod_{i=1}^{N} \prod_{k\in\mathcal{F}_{i,t}} p\big(\boldsymbol{o}_{i,k,t}^{(\mathrm{A2T})}|\boldsymbol{s}_{i,t}^{(\mathrm{A})}, \boldsymbol{s}_{k,t}^{(\mathrm{T})}\big)\,. \end{aligned} \tag{8}$$

For notation purposes, we will denote the likelihood function also as $O(\boldsymbol{o}_t|\boldsymbol{s}_t) \triangleq p(\boldsymbol{o}_t|\boldsymbol{s}_t)$. In case the dynamic and measurements models in (1) and (5), respectively, are linear and with a Gaussian noise, the state estimate in (6) reduces to a Kalman filter (KF) as described in [32], [46], with efficient resolution in matrix form.

B. Limitations of Bayesian ICP Methods

The centralized ICP approach is impractical for extensive networks due to the following major limitations: the single central computing unit representing a point of failure, and its computational complexity growing cubically with the number of vehicles and passive objects [32]. To overcome such limitations, distributed or consensus-based ICP algorithms have been studied in the past [34]. However, their convergence to the centralized solution is guaranteed only in acyclic (i.e., tree-structured) factor graphs. Moreover, even in case of convergence, the result would be optimal only with Gaussian and linear models (i.e., in (1) and (5)). In all the other cases, optimality is not guaranteed. In Fig. 2 we summarized all cases and highlighted those where improvements could be provided by new data-driven designs. We point out that in real-world dynamics, the factor graph is usually not acyclic and the models are typically neither Gaussian nor linear.

The aim of this paper is to address the gap by proposing a new decentralized data-driven solution to the ICP problem suited for non-linear non-Gaussian models, overcoming the limits of parametric Bayesian implementations based on EKF or particle filter (PF) highlighted in Fig. 2. The proposed distributed method also incorporates a data-driven optimization of the cooperation graph by making the agents actively and opportunistically select the cooperating neighbors so as to minimize the communication signaling. In particular, to address the limitations of conventional ICP solutions, we adopt neural networks (NNs)-based models, which are able to learn whatever non-linear function is hidden in the data thanks to the universal approximation theorem. Specifically, a RNN learns the non-linear motion and measurement models, whereas a multi-layer perceptron (MLP) learns the non-linear relation between link activation and state estimate. Moreover, NNs have proven effective even in non-Gaussian settings [53], given their ability to model complex probability distributions without assuming any specific form. The centralized ICP method reviewed in this section will be used as a benchmark to assess the proposed method.

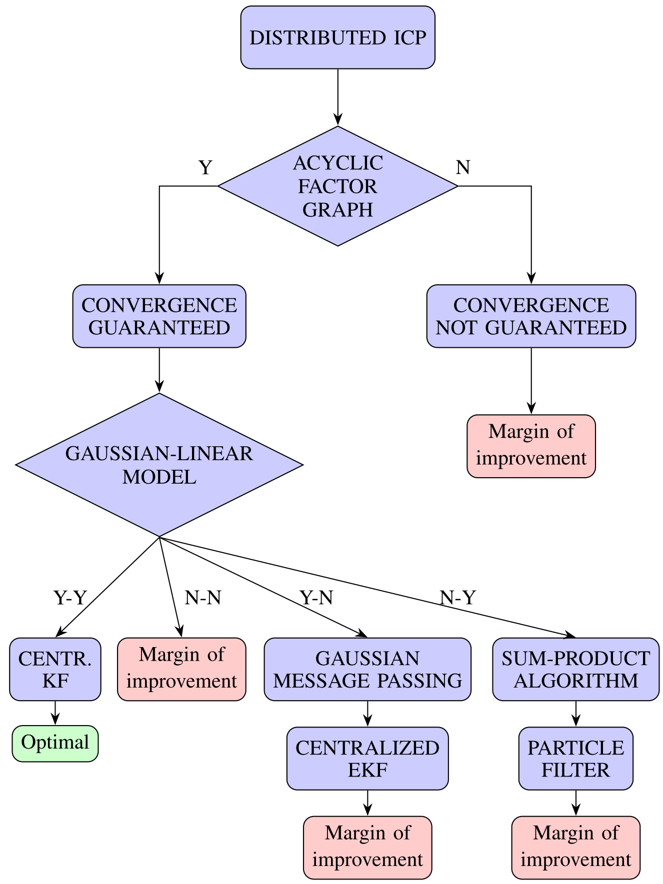


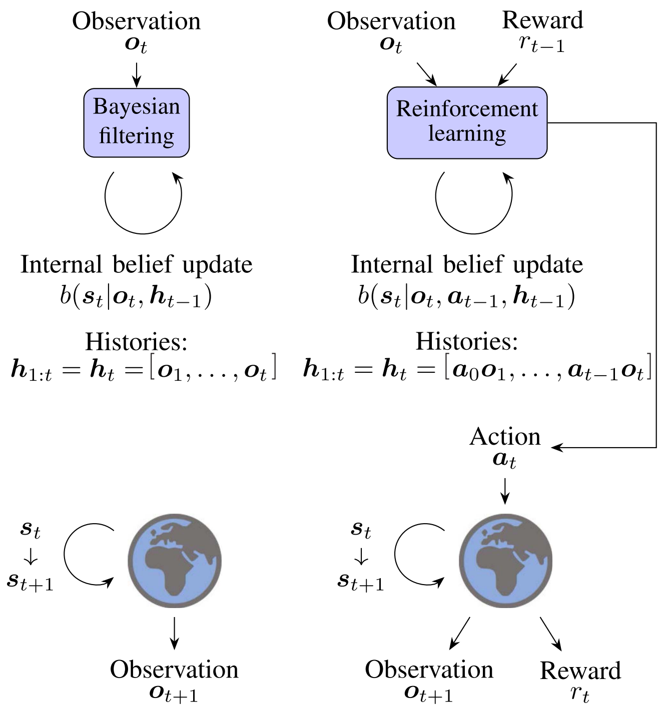
Fig. 2. Convergence conditions and optimality in ICP methods.

IV. MARL for Cooperative Positioning

In this section, we first introduce the MARL framework (Section IV-A) that will be used later for the design of the ICP-MAPPO solution (Section IV-B). The ICP-MAPPO execution and training schemes are reported in Sections IV-C and IV-D, respectively.

A. MARL Framework

We model the cooperative MAS as a finite-horizon Dec-POMDP [75] defined by the tuple $\langle \mathcal{V}, \mathcal{S}, \mathcal{A}, T_0, T, \mathcal{O}, O, R, \gamma, H\rangle$. We recall that the set $\mathcal{V}$ refers to the cooperative agents, while the sets $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. T_0 is the initial state distribution at time $t = 0$, while $T(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_t) \triangleq p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_t)$ is the state transition PDF that, differently from the Bayesian-filtering system model in Section II, now also includes the joint action realization $\boldsymbol{a}_t = [\boldsymbol{a}_{i,t}]_{i\in\mathcal{V}} \in \mathcal{A}$ and the joint state $\boldsymbol{s}_t \in \mathcal{S}$.



(a) Bayesian filtering (b) Reinforcement learning

Fig. 3. Comparison between Bayesian filtering and RL.

At each time t, the agents receive the joint observations or measurements $\boldsymbol{o}_t \in \mathcal{O}$ which are sampled from the distribution $O(\boldsymbol{o}_t|\boldsymbol{a}_{t-1}, \boldsymbol{s}_t) \triangleq p(\boldsymbol{o}_t|\boldsymbol{a}_{t-1}, \boldsymbol{s}_t)$. Note that here, (8) is also function of the previous joint action of the agents $\boldsymbol{a}_{t-1}$, thus generalizing the concept of Bayesian-filtering. $R(\mathsf{s}_t, \mathsf{a}_t) = \mathsf{r}_t \in \mathbb{R}$ denotes the instantaneous shared reward at time t obtained from the reward function R, while $\gamma \in [0, 1)$ and H are the discount factor and time horizon of each episode, respectively.

Since the rewards and states are not directly observable by the agents, the system is a partially observable MDP, where each agent i needs to keep track of the so-called *histories* defined as $\boldsymbol{h}_{i,1:t} = \boldsymbol{h}_{i,t} = [(\boldsymbol{a}_{i,t'-1}, \boldsymbol{o}_{i,t'})]_{t'=1}^{t}$. Note that the histories are a generalization of the aggregated measurements up to time t in (6). Given a new observation $\boldsymbol{o}_{i,t}$, the state estimates $\widehat{\boldsymbol{s}}_{i,t}$ are produced by MMSE criterion from the belief PDF $b_{\boldsymbol{\psi}}(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{i,t}, \boldsymbol{a}_{i,t-1}, \boldsymbol{h}_{i,t-1}) = p_{\boldsymbol{\psi}}(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{i,t}, \boldsymbol{a}_{i,t-1}, \boldsymbol{h}_{i,t-1})$ parameterized by $\boldsymbol{\psi}$. Moreover, agents adopt a policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}_{i,t}) = p_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}_{i,t})$ defined by $\boldsymbol{\theta}$ to obtain the action $\boldsymbol{a}_{i,t}$ from histories $\boldsymbol{h}_{i,t}$. A full comparison between Bayesian filtering and RL (i.e., its generalized version) can be found in Fig. 3. By defining the *reward-to-go* $\mathsf{R}_t = \sum_{t'=t}^{H-1} \gamma^{t'-t} \mathsf{r}_{t'}$ as the cumulative discounted reward from time t to the end of the episode, the objective of the MARL problem is to maximize, over the policy π, the expected cumulative discounted reward from the beginning of the episode

$$\max_{\pi} J(\pi) = \max_{\pi} \mathbb{E}\{\mathsf{R}_0\} \tag{9}$$

which usually translates into optimizing the parameters of the policy as $\boldsymbol{\theta}^* = \mathrm{argmax}_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})$, with $\pi^*_{\boldsymbol{\theta}}$ representing the optimal policy.

B. MARL Solution to the ICP Problem

In standard Dec-POMDP, each agent only knows its local actions and observations, thus resulting in possible non-stationary learning problems from each agent's perspective [96]. By training independent learners to optimize the team reward (i.e., concurrent learning), we induce a change in the dynamics of the environment as teammates continuously adapt their behaviours throughout learning. On the contrary, whenever a fully connected graph with communications is present, the Dec-POMDP collapses to a centralized POMDP, resulting in higher complexity and communication inefficiencies [89], [97], exactly as in centralized ICP. To solve the issues of independent and centralized training-execution, the state-of-the-art works exploit the so called centralized-training and decentralized-execution paradigm. This framework permits to learn the policies in a centralized way and then deploy them in the network graph for decentralized execution [85], [87], [98].

While this approach solves the problem in standard MARL algorithms, in the context of ICP, having access to the neighbors' measurements would allow the positioning accuracy to be significantly improved. Indeed, the objective of ICP is to minimize over the belief b the error on the state estimate as

$$\min_{b} J(b) = \min_{b} \mathbb{E}\bigg\{ \sum_{t} \big\|\mathsf{s}_t - \widehat{\boldsymbol{s}}_t\big\|_2^2 \bigg\}. \tag{10}$$

Therefore, we here propose to define as actions the agent's selection of the communication links to the neighbors to cooperate with. This allows to optimize the communication efficiency with respect to the centralized solution. Formally, we define the following Dec-POMDP:

1) Agents: The agent is identified by vehicle $i \in \mathcal{V}$ that composes the connected network.

2) Actions: The action of agent i at time t is $\mathsf{a}_{i,t} = [\mathsf{a}_{i,j,t}]_{j=1}^{N}$, where $\mathsf{a}_{i,j,t} \in \{0, 1\}$ represents the Boolean decision of agent i to communicate with agent j.

3) States: Only the states of the vehicles $\mathsf{s}_t^{(\mathrm{A})}$ are considered, while the target states $\mathsf{s}_t^{(\mathrm{T})}$ are implicitly learned by the NNs through the hidden features. Indeed, the system does not output or keep track of the states of the targets, since they are not needed as in the ICP Bayesian filtering formulation. In other words, the ICP-MAPPO model just outputs the predicted states of the agents, while the targets' states are contained in the hidden space, i.e., histories. Therefore, from now on, we indicate with s_t the state of the agents $\mathsf{s}_t^{(\mathrm{A})}$.

4) Observations: GNSS, A2A, and A2T measurements described in Section II are the observations used in the Dec-POMDP modeling, as they are the only output returned by the world at inference time.

During the centralized training, the agents learn the relation between histories-actions, i.e., policy optimization, and histories-states, i.e., belief optimization, while having access to the full observable state s_t and measurements o_t. Conversely,

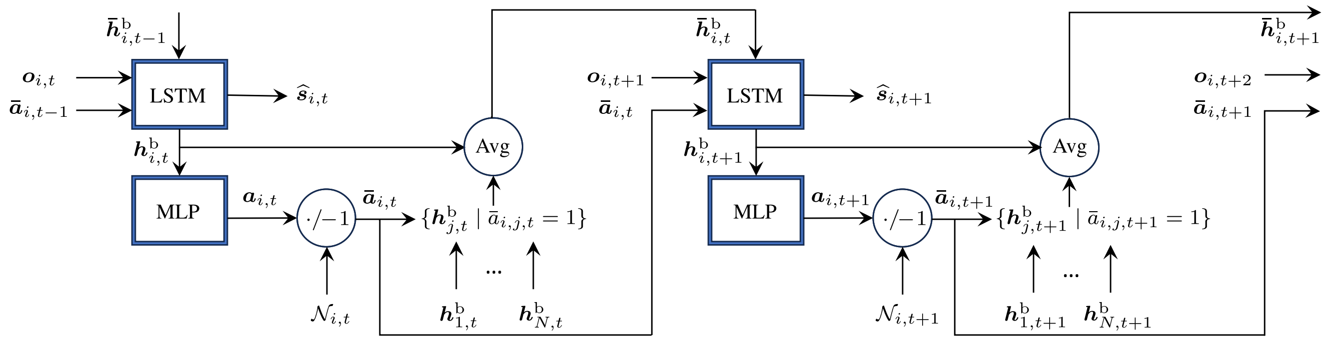


Fig. 4. Dynamic-decentralized-execution scheme of the proposed ICP-MAPPO algorithm.

during the decentralized execution, the agents decide how to modify the network graph to achieve the best trade-off between positioning accuracy and communication efficiency. We call this approach *centralized-training and dynamic-decentralized-execution*, as during execution, according to the agents' actions, the coordination graph may vary, passing from fully-connected to fully-decentralized according to the agent's decisions.

C. ICP-MAPPO Execution Scheme

For belief and action prediction, we propose to employ long short-term memory (LSTM) and MLP, respectively. In Fig. 4, we show a compact representation of the execution within each agent. In particular, the NN functions are defined as

$$\widehat{\boldsymbol{s}}_{i,t}, \boldsymbol{h}^{\mathrm{b}}_{i,t} = b_{\boldsymbol{\psi}}(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{i,t}, \bar{\boldsymbol{a}}_{i,t-1}, \bar{\boldsymbol{h}}^{\mathrm{b}}_{i,t-1}) \tag{11}$$

$$\boldsymbol{a}_{i,t} \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}^{\mathrm{b}}_{i,t}) \tag{12}$$

where $\boldsymbol{o}_{i,t}$ is the ordered vector of all measurements of agent i at time t defined as in Section II, $\bar{\boldsymbol{a}}_{i,t} = [\bar{a}_{i,j,t}]_{j=1}^{N}$ includes the sampled actions from the policy distribution adjusted with the feasibility of the network connectivity as

$$\bar{a}_{i,j,t} = \begin{cases} a_{i,j,t} & \text{if } \ j \in \mathcal{N}_{i,t} \\ -1 & \text{otherwise} \end{cases} \tag{13}$$

and $\bar{\boldsymbol{h}}^{\mathrm{b}}_{i,t}$ are the hidden features of the belief LSTM which contain a compressed representation of the histories of agent i and all selected neighbors at the previous timestep

$$\bar{\boldsymbol{h}}^{\mathrm{b}}_{i,t} = \frac{\boldsymbol{h}^{\mathrm{b}}_{i,t} + \sum_{j\in\mathcal{V}} \boldsymbol{h}^{\mathrm{b}}_{j,t}\, \mathbb{1}(\bar{a}_{i,j,t} == 1)}{1 + \sum_{j\in\mathcal{V}} \mathbb{1}(\bar{a}_{i,j,t} == 1)} \tag{14}$$

where $\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise. We point out that the hidden features $\boldsymbol{h}^{\mathrm{b}}_{i,t}$ include not only past actions and measurements but also the implicit state estimates of the targets $\widehat{\boldsymbol{s}}^{(\mathrm{T})}_{t}$, which are never explicitly predicted by the system for output space complexity reduction.

The key rationale behind the proposed execution scheme is the following. We employ the average operation in (14) to avoid gradient divergence over the timesteps. Furthermore, the action decision at time t in (12) is mainly based on the previous timestep information $\bar{\boldsymbol{h}}^{\mathrm{b}}_{i,t-1}$, as there is no way for agent i to know a priori the measurements of its neighbors $\boldsymbol{h}^{\mathrm{b}}_{j,t}, \forall j \in \mathcal{V}$, in order to activate the communications between them. Moreover, the actions $\bar{\boldsymbol{a}}_{i,t}$ are given as input to the belief LSTM for two main reasons. First, the information about which agents were selected for measurements fusion is necessary to coherently predict the state estimate. Second, the negative action values imposed by the lack of possible connectivity permit each agent to implicitly learn its index or identification. In this way, the scalable and efficient parameter sharing approach for training one single NN [89], instead of agent-specific NNs, can be combined with agent differentiation by index learning.

D. ICP-MAPPO Training Scheme

For the reward definition, we propose to use a function that, looking at the future timestep, rewards the actions that gave a predetermined improvement β on the positioning accuracy. In other words, each agent i tries to answer the following question: if I had chosen agent j' instead of agent j, would the performances have improved? This is formalized as

$$r_t = \begin{cases} -1 & \text{if } \ \left\|\boldsymbol{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2 - \left\|\boldsymbol{s}_{t+1} - \widehat{\boldsymbol{s}}_{t+1}\right\|_2^2 \leq -\beta \\ +1 & \text{if } \ \left\|\boldsymbol{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2 - \left\|\boldsymbol{s}_{t+1} - \widehat{\boldsymbol{s}}_{t+1}\right\|_2^2 > \beta \\ +2 & \text{if } \ -\beta < \left\|\boldsymbol{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2 - \left\|\boldsymbol{s}_{t+1} - \widehat{\boldsymbol{s}}_{t+1}\right\|_2^2 \leq \beta \end{cases} \tag{15}$$

where β is a hyper-parameter which regulates the improvement step. At the beginning of the learning, if the improvement is negative and bigger than β, the reward is negative as the actual agent selection worsen the positioning accuracy. On the other hand, if the improvement is positive and greater than β, the reward is +1. Finally, when the learning starts converging and the improvements become smaller, we introduce a long-term reward of +2. Note that, while in conventional Dec-POMDPs the reward directly depends on the actions, in the proposed system the effect of the actions' choice can be assessed only at the next timestamp and only by measuring the positioning error.

Regarding the selection of MARL algorithm, we opted for PO over Q-learning-based methods. This is because Q-learning algorithms combined with DL have no guarantees of convergence and retain a lot of bias (i.e., inaccurate state-action value or Q-value). On the contrary, PO algorithms retain very low bias since they directly optimize the objective function in (9) and have been proven to outperform Q-learning methods in MARL systems [87]. Moreover, while off-policy RL algorithms use historical data to learn the policy, in the context of CP, where state estimation is crucial, it is essential to utilize the most up-to-date policy available since the action sampling (i.e., radio link activation) directly influences the positioning performances. Despite PO algorithms having an intrinsic high variance, i.e., they require a lot of samples to converge, this can be mitigated by the learning of the value function, either $V^\pi(\boldsymbol{s}_t)$ or $Q^\pi(\boldsymbol{s}_t, \boldsymbol{a}_t)$, which estimates the long-term reward given a specific state or state-action pair, respectively. Specifically, we employ the state value function defined as

$$\begin{aligned} V^\pi(\boldsymbol{s}_t) &= \mathbb{E}\{\mathsf{R}_t | \mathsf{s}_t = \boldsymbol{s}_t\} \\ &= \mathbb{E}_{\mathbf{a}_t \sim \pi, \mathbf{s}_{t+1} \sim T}\Big\{R(\boldsymbol{s}_t, \mathbf{a}_t) + \gamma V^\pi(\mathbf{s}_{t+1})\Big\}. \end{aligned} \quad (16)$$

Usually, $V^\pi(\boldsymbol{s}_t)$ cannot be directly computed due to the curse of dimensionality and thus it is estimated by an additional NN $\widehat{V}_{\boldsymbol{\phi}}(\boldsymbol{s}_t) = V_{\boldsymbol{\phi}}(\boldsymbol{s}_t)$, with parameters $\boldsymbol{\phi}$ which are only employed during training.

In standard single-agent RL frameworks, the policy optimization problem is usually defined with the introduction of trajectories $\boldsymbol{\tau} = (\mathbf{s}_0, \mathbf{a}_0, \dots, \mathbf{s}_H, \mathbf{a}_H)$ by maximizing

$$\begin{aligned} J(\pi_{\boldsymbol{\theta}}) &= \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})}\left\{\widetilde{R}(\boldsymbol{\tau})\right\} \\ &= \sum_{t=0}^{H} \mathbb{E}_{\mathbf{s}_t \sim p(\boldsymbol{s}_t|\pi_{\boldsymbol{\theta}}), \mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\Big\{\gamma^t R(\mathbf{s}_t, \mathbf{a}_t)\Big\} \end{aligned} \quad (17)$$

where $\widetilde{R}(\boldsymbol{\tau}) = \mathsf{R}_0$ is the reward of trajectory $\boldsymbol{\tau}$, $p(\boldsymbol{\tau}|\pi) = T_0 \prod_{t=0}^{H-1} T(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)\, \pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ is the PDF of an H-step trajectory, and $p(\boldsymbol{s}_t|\pi)$ is the state marginal of the trajectory distribution induced by policy π. Standard REINFORCE PO algorithms [99] update the policy parameters in (17) in the direction of $\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})$, which can be written as (see Appendix A)

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim p(\boldsymbol{s}_t, \boldsymbol{a}_t|\pi_{\boldsymbol{\theta}})}\Big\{\sum_{t=0}^{H-1} \nabla_{\boldsymbol{\theta}} \log\big(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\big) A_t\Big\} \quad (18)$$

where $p(\boldsymbol{s}_t, \boldsymbol{a}_t|\pi_{\boldsymbol{\theta}})$ is the state-action marginal of the trajectory distribution induced by policy π and $A_t = A_t(\mathbf{s}_t, \mathbf{a}_t)$ is the generic advantage function at time t [100], which quantifies the convenience of taking a specific action $\mathbf{a}_t$ in a given state $\mathbf{s}_t$, compared to the average action's expected return for that state.

During successive optimization steps of (18) within the same trajectory, where the objective is to maintain proximity between new and old policy parameters, even minor variations in the NN weights can lead to significant differences in performance. Consequently, a single unfavorable optimization step can drastically deteriorate the policy's effectiveness. Recent state-of-the-art methods, e.g., trust region policy optimization (TRPO) [101] and proximal policy optimization (PPO) [102], tried to solve this problem by taking the largest gradient step size possible to improve performance, while maintaining constraints on how close the new and old policies (i.e., $\pi_{\boldsymbol{\theta}_{\text{old}}}$ at previous train step) are allowed to be. The constraint in TRPO is enforced by Kullback–Leibler (KL) divergence and the parameters are obtained by maximizing the *surrogate* objective function as

$$\begin{aligned} \boldsymbol{\theta} &= \operatorname{argmax}_{\boldsymbol{\theta}}\ \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim p(\boldsymbol{s}_t, \boldsymbol{a}_t|\pi_{\boldsymbol{\theta}})}\Big\{\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} A_t(\mathbf{s}_t, \mathbf{a}_t)\Big\} \\ &\text{s.t. } \mathbb{E}_{\mathbf{s}_t \sim p(\boldsymbol{s}_t|\pi_{\boldsymbol{\theta}})}\Big\{D_{\text{KL}}\left(\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t)\big\|\pi_{\boldsymbol{\theta}_{\text{old}}}(\cdot|\boldsymbol{s}_t)\right)\Big\} \leq \epsilon \end{aligned} \quad (19)$$

which resulted in a second-order optimization method. On the contrary, PPO and its recent multi-agent version MAPPO use a much more efficient first-order method that exploits clipping to remove incentives for the new policy to get far from the old policy.

In this paper, we adopt three loss functions: $L(\boldsymbol{\phi})$ and $L(\boldsymbol{\theta})$ derived from the MAPPO scheme to train the state-value and policy NNs, respectively, and $L(\boldsymbol{\psi})$ to train the belief NN. $\pi_{\boldsymbol{\theta}}$ and $V_{\boldsymbol{\phi}}$ are called *actor* and *critic*, respectively, since the actor is responsible for selecting actions based on the current policy, and the critic evaluates the quality of these actions by estimating the value function. In Dec-POMDP, the critic $V_{\boldsymbol{\phi}}$ is also dependent on the history of action-observation pairs and thus it is usually modelled with a RNN as

$$\widehat{V}_{\boldsymbol{\phi}}(\boldsymbol{s}_{i,t}, \boldsymbol{h}^{\text{V}}_{i,t-1}), \boldsymbol{h}^{\text{V}}_{i,t} = V_{\boldsymbol{\phi}}(\boldsymbol{s}_{i,t}, \boldsymbol{h}^{\text{V}}_{i,t-1}) \quad (20)$$

where $\boldsymbol{h}^{\text{V}}_{i,t}$ are the hidden features of the critic. Given a trajectory of length L_τ (subset of the horizon length H), $L(\boldsymbol{\phi})$ is defined to perform regression on the rewards-to-go as

$$\begin{aligned} L(\boldsymbol{\phi}) = \frac{1}{N L_\tau} \sum_{i \in \mathcal{V}} \sum_{\ell=1}^{L_\tau} \Bigg\{ &\max\Bigg(\Big[\widehat{V}_{\boldsymbol{\phi}}(\boldsymbol{s}_{i,\ell}, \boldsymbol{h}^{\text{V}}_{i,\ell}) - R_\ell\Big]^2, \\ &\Big[\text{clip}\left(\widehat{V}_{\boldsymbol{\phi}}(\boldsymbol{s}_{i,\ell}, \boldsymbol{h}^{\text{V}}_{i,\ell-1}), \widehat{V}_{\boldsymbol{\phi}_{\text{old}}}(\boldsymbol{s}_{i,\ell}, \boldsymbol{h}^{\text{V}}_{i,\ell-1}), \epsilon\right) - R_\ell\Big]^2\Bigg)\Bigg\} \end{aligned} \quad (21)$$

where the clip prevents the value function from radically changing between iterations, and it is defined as

$$\text{clip}(A, B, \epsilon) = \min\Big(\max(A, B - \epsilon), B + \epsilon\Big) \quad (22)$$

where ϵ is the clip coefficient.

The actor $\pi_{\boldsymbol{\theta}}$ is also trained with clipping to discard the KL constraint in (19) by minimizing

$$\begin{aligned} L(\boldsymbol{\theta}) = -\frac{1}{N L_\tau} \sum_{i \in \mathcal{V}} \sum_{\ell=1}^{L_\tau} \Bigg\{ &\min\Bigg(\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,\ell}|\boldsymbol{h}^{\text{b}}_{i,\ell})}{\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{a}_{i,\ell}|\boldsymbol{h}^{\text{b}}_{i,\ell})} \widehat{A}_{i,\ell}, \\ &\text{clip}\Big(\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,\ell}|\boldsymbol{h}^{\text{b}}_{i,\ell})}{\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{a}_{i,\ell}|\boldsymbol{h}^{\text{b}}_{i,\ell})}, 1, \epsilon\Big) \widehat{A}_{i,\ell}\Bigg) + \alpha S\left(\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{h}^{\text{b}}_{i,\ell})\right)\Bigg\} \end{aligned} \quad (23)$$

where $\widehat{A}_{i,\ell} = R_\ell - \widehat{V}_{\boldsymbol{\phi}_{\text{old}}}(\boldsymbol{s}_{i,\ell}, \boldsymbol{h}^{\text{V}}_{i,\ell-1})$ is the advantage function estimate, $S(p_{\mathsf{x}}) = \mathbb{E}_{\mathsf{x} \sim p_{\mathsf{x}}}\{-\log(p_{\mathsf{x}}(x))\}$ is the entropy function which encourages the exploration by inducing stochastic policies, and α is the temperature hyper-parameter which balances

Algorithm 1: Implicit Cooperative Positioning Multi-Agent Proximal Policy Optimization (ICP-MAPPO).

Input: actor, critic and belief parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{old}}$, $\boldsymbol{\phi} = \boldsymbol{\phi}_{\text{old}}$, and $\boldsymbol{\psi}$.
for each training step $n = 1, 2, \ldots, N_{\text{step}}$ **do**
  Initialize empty batch $\mathcal{B} = \{\}$ and trajectory $\boldsymbol{\tau} = [\,]$
  Initialize histories $\boldsymbol{h}^{\text{V}}_{i,0}$ and $\boldsymbol{h}^{\text{b}}_{i,1}$ for critic and beliefs
  Initialize state estimate $\widehat{\boldsymbol{s}}_0$
  for $t = 1, 2, \ldots, H$ **do**
    for all agents $i \in \mathcal{V}$ **in parallel do**
      Sample action $\boldsymbol{a}_{i,t} \sim \pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}^{\text{b}}_{i,t})$
      Send $\boldsymbol{h}^{\text{b}}_{i,t}$ and receive $\boldsymbol{h}^{\text{b}}_{j,t}$ $\forall j \in \mathcal{N}_{i,t}$
      Get value estimate $\widehat{V}_{\boldsymbol{\phi}_{\text{old}}}(\boldsymbol{s}_{i,t}, \boldsymbol{h}^{\text{V}}_{i,t-1})$ with (20)
      Compute $\bar{\boldsymbol{a}}_{i,t}$ and $\bar{\boldsymbol{h}}^{\text{b}}_{i,t}$ with (13) and (14)
      Observe $\boldsymbol{s}_{i,t+1}, \boldsymbol{o}_{i,t+1}$
      Get state estimate $\widehat{\boldsymbol{s}}_{i,t+1}$ with (11)
    end for
    Observe r_t and store $\boldsymbol{\tau}_t$ in $\boldsymbol{\tau}$
  end for
  Compute advantage estimate $\widehat{A}_{i,t}$ $\forall t$ and agent i on $\boldsymbol{\tau}$
  Compute reward-to-go R_t for each $\forall t$ on $\boldsymbol{\tau}$
  Split trajectory τ into chunks of length L_τ
  for each $\ell = 0, 1, \ldots, \lfloor H/L_\tau \rfloor$ **do**
    $\mathcal{B} = \mathcal{B} \cup \{\boldsymbol{\tau}_t, \widehat{A}_t, R_t\}_{t=\ell}^{\ell+L_\tau}$
    Adam update of $\boldsymbol{\psi}$ on $L(\boldsymbol{\psi})$ with data $\{\boldsymbol{\tau}_t\}_{t=\ell}^{\ell+L_\tau}$
  end for
  for each mini-batch **do**
    Sample $\{\boldsymbol{\tau}_\ell\}_{\ell=1}^{L_\tau} \sim \mathcal{B}$
    Adam update of $\boldsymbol{\theta}$ on $L(\boldsymbol{\theta})$ with data $\{\boldsymbol{\tau}_\ell\}_{\ell=1}^{L_\tau}$
    Adam update of $\boldsymbol{\phi}$ on $L(\boldsymbol{\phi})$ with data $\{\boldsymbol{\tau}_\ell\}_{\ell=1}^{L_\tau}$
  end for
  $\boldsymbol{\theta}_{\text{old}} \leftarrow \boldsymbol{\theta}$, $\boldsymbol{\phi}_{\text{old}} \leftarrow \boldsymbol{\phi}$
end for

the trade-off between exploiting the best actions and exploring new actions. Finally, the beliefs $b_{\boldsymbol{\psi}}$ adopt a MSE loss function to minimize $J(b)$ in (10) as

$$L(\boldsymbol{\psi}) = \frac{1}{NL_\tau} \sum_{i \in \mathcal{V}} \sum_{\ell=1}^{L_\tau} \left\| \widehat{\boldsymbol{s}}_{i,t} - \boldsymbol{s}_{i,t} \right\|_2^2. \quad (24)$$

All the NNs are trained with maximum likelihood estimation (MLE) criterion. However, while $b_{\boldsymbol{\psi}}(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{i,t}, \bar{\boldsymbol{a}}_{i,t-1}, \bar{\boldsymbol{h}}^{\text{b}}_{i,t-1})$ directly outputs $\widehat{\boldsymbol{s}}_{i,t}$, $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}^{\text{b}}_{i,t})$ predicts the probability of communication among agents through sigmoid activation functions, from which actions $\boldsymbol{a}_{i,t}$ are sampled. The full training algorithm can be found in Algorithm 1, where we defined a transition as $\boldsymbol{\tau}_t = (\boldsymbol{s}_t, \boldsymbol{o}_t, \boldsymbol{h}^{\text{b}}_t, \bar{\boldsymbol{h}}^{\text{b}}_t, \boldsymbol{h}^{\text{V}}_t, \boldsymbol{a}_t, \bar{\boldsymbol{a}}_t, r_t, \boldsymbol{s}_{t+1}, \boldsymbol{o}_{t+1}, \widehat{\boldsymbol{s}}_{t+1})$. Since our approach combines the usage of passive targets to improve the position estimate and MAPPO MARL to perform an efficient agent selection, we call this algorithm ICP-MAPPO.

The main characteristics of ICP-MAPPO are the following. ICP-MAPPO is a low-bias on-policy algorithm since the data used to train the agents are collected from the policy currently being learned or improved. For value regression, we adopted a centralized value function that takes as input extra global information (i.e., the states) not present in the agent's local observation to accurately estimate the values state. The beliefs are computed as in model-based value estimation (MBVE) RL [103], [104], leveraging the learned dynamics to predict the state estimate. This additionally reduces the variance of the PO method without introducing additional biases by avoiding performing rollouts [105]. Finally, as opposed to conventional MARL algorithms, the rewards are not directly dependent on the action, but only implicitly through the beliefs of the next timestep. This permits to effectively decouple the evaluation of actions based on the improvement of state predictions rather than immediate outcomes, focusing on long-term strategic benefits rather than short-term gains.

V. Simulation Experiments

In this section, we first introduce the scenario and the training procedures, and then we describe the baseline methods, and the main simulation results.

A. Simulation Setup

To evaluate the performances of the proposed ICP-MAPPO algorithm, we simulate a C-ITS scenario with the CARLA software [94] in an urban map (i.e., *Town02* of CARLA) that spans an area of 200×200 m^2. Fig. 1 shows a bird-eye-view representation of the map. CARLA takes into account inter-vehicle dynamics, such as acceleration, braking behavior, and collision physics, as well as communication constraints given by the environment. Within the area, 20 CAVs move for 1500 timesteps sampled every 0.2 s, while 72 fixed objects (poles) are detected by the vehicles if in line-of-sight (LoS) and within a sensing range of 70 m. The same coverage area applies to A2A measurements. For the communications, we only consider the direct LoS path, as if the vehicles were equipped with LIDAR technology that could be blocked by obstacles such as buildings or other vehicles. The absolute driving speed adopted in the testing scenario ranges from 0 to about 60 km/h, with a mean and standard deviation speed of 0.2 km/h and 14 km/h, respectively. We point out that the motion models of the vehicles are not linear and that the factor graph to solve the distributed ICP method contains cycles. For the GNSS, A2A, and A2T observations, measurement errors are simulated as additive independent Gaussian noises with standard deviations of 2 m each.

For the training and testing of the ICP-MAPPO algorithm, we create two different simulations each composed of $H = 1500$ timesteps. Model training is performed over $N_{\text{step}} = 2000$ episodes (or training steps), each characterized by a different realization of the measurements. For testing, 40 Monte Carlo (MC) evaluations are considered, unless otherwise specified. During training, we adopt a trajectory length $L_\tau = H/2$ to use at most 2 mini-batches, as suggested by [87], [106]. The entropy, reward and clipping coefficients have been chosen to be $\alpha = 0.01$, $\beta = 0.05$ and $\epsilon = 0.2$, respectively. Note that $\beta = 0.05$ would correspond to an improvement step of the reward function of 5 cm in a non-standardized state scenario. The discount factor is

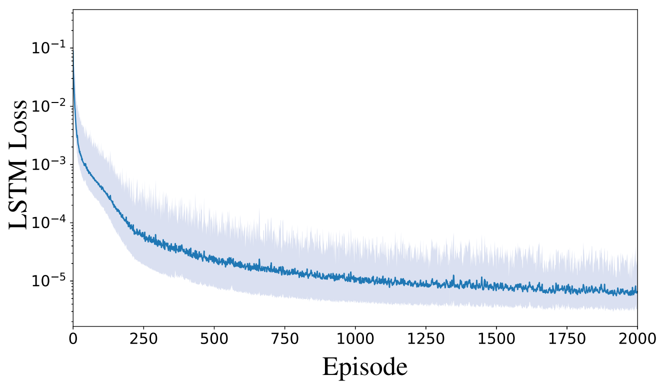


Fig. 5. Belief LSTM loss varying the number of training episodes.

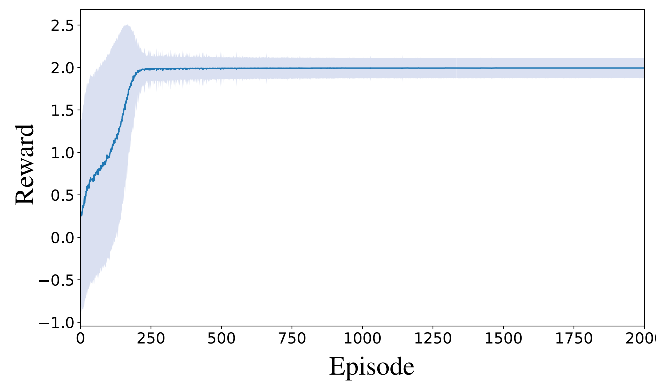


Fig. 6. Achieved reward varying the number of training episodes.

$\gamma = 0.99$, while the Adam [107] learning rate is $\mu = 10^{-5}$ with standard hyper-parameters.

Regarding the NN architectures, we adopt a critic network with three layers: a fully-connected (FC) linear layer with 256 neurons, a gated recurrent unit (GRU) with hidden size of 256 and a final FC linear layer. The actor is an MLP with two hidden linear layers of [128, 64] neurons and rectified linear unit (ReLU) activation functions, and an output layer with sigmoid activation function. Lastly, the belief network employs two bidirectional LSTM layers of 256 hidden neurons each and ReLU activation functions, followed by a Maxout unit with 128 output features and two linear layers of [64, 32] neurons.

B. Computational Complexity and Latency

To access the real-time processing capabilities of the proposed method in fulfilling the CAVs requirements on latency, we here investigate the computational complexities and communication delays of the proposed ICP-MAPPO solution with respect to the ICP algorithm. We specify that the number of floating point operations (FLOPs) for $V_{\boldsymbol{\phi}}$, $\pi_{\boldsymbol{\theta}}$ and $b_{\boldsymbol{\psi}}$ are $0.82 \cdot 10^6$, $0.54 \cdot 10^6$, and $11.3 \cdot 10^6$, respectively. For comparison, the computational complexity of particle-based ICP methods is estimated with $O(N_{\text{mp}} \cdot N \cdot K \cdot N_{\text{p}})$, where N_{mp} and N_{p} are the number of message passing iterations and particles, respectively. The experiments are performed on a workstation machine with Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz, 96 GB RAM, and a Quadro RTX 6000 24 GB GPU, capable of achieving about $16.3 \cdot 10^{12}$ floating point operations per second (FLOPS) with just CPU performances. This implies a maximum latency for sample-inference of around 1 μs, which is expected to be truthful and accurate since the computational capabilities of CAVs are planned to far exceed our workstation capabilities with more than $4 \cdot 10^{15}$ FLOPS for L5 SAE level [108].

When considering the communication delays with a hidden LSTM size of 256 bytes for ICP-MAPPO and about $N_{\text{mp}} = 1000$ particles (each with 2 bytes for 2D position and 1 B for the weight) in the ICP method, the data transmission would require approximately 1 and 10 packets, respectively. This estimate is based on 5G vehicle-to-vehicle (V2V) communications with a typical packet size of 300 bytes. Two communication scenarios are possible: direct V2V [109] or vehicle-to-network-to-vehicle (V2N2V) [110] when under cellular coverage. For direct V2V communication, the end-to-end (E2E) packet latency is around 1 ms [109], resulting in 10 ms for ICP and 1ms for ICP-MAPPO. In the V2N2V case, assuming the distances and scenarios described in [110], the E2E packet latency is around 4ms, resulting in 40 ms for ICP and 4ms for ICP-MAPPO. We note that the ICP E2E communication delay exceeds the 5 ms latency requirements of fully CAVs [111] in both scenarios, especially if a message passing procedure with multiple belief exchanges is considered. On the contrary, the ICP-MAPPO method meets the stringent latency requirements needed for fully CAVs.

C. Baseline Methods

As benchmark algorithms, we consider the following implementations:

1) KF-GNSS: Non-cooperative single-agent GNSS-based KF only using GNSS observations and perfect knowledge of the measurement standard deviation $\sigma^{(\text{GNSS})} = 2$ m. For the motion dynamics (1), we adopt a constant velocity model with standard deviation of the Gaussian-distributed velocity driving process calibrated on the data and equal to 0.5 m/s^2.

2) ICP: Centralized ICP method from [32] with known A2A and A2T standard deviations, i.e., $\sigma^{(\text{A2A})} = \sigma^{(\text{A2T})} = 2$ m, and same motion model as for the KF-GNSS. Note that the use of the exact measurement statistics in generation and tracking allows to obtain the optimal performance (i.e. with no errors due to mis-modeling). Here the network of agents is fully-connected, i.e., all the agents share the same measurements.

3) Ego ICP-MAPPO: Proposed ICP-MAPPO method, with no-cooperation, i.e., only comprising the belief LSTM and imposing no connectivity with other agents, i.e., $\bar{a}_{i,j,t} = -1 \;\; \forall t \in \{0, \ldots, H-1\}, i \in \mathcal{V}, j \in \mathcal{N}_{i,t}$. In this way, each agent has to rely just on its measurements without performing aggregation of the neighbors' hidden features.

D. Results

1) Training Performances: In the first assessment, we aim at verifying the convergence of the proposed ICP-MAPPO algorithm during the training episodes. In Figs. 5, 6 and 7, we report the mean belief LSTM loss, reward, and state value function, respectively, along with the 5–95 percentile as error bounds.

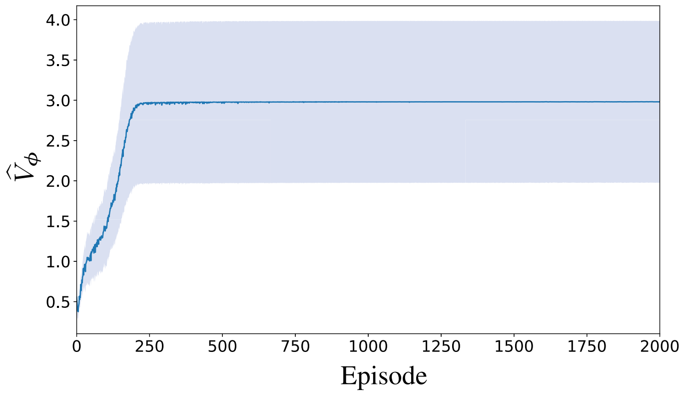
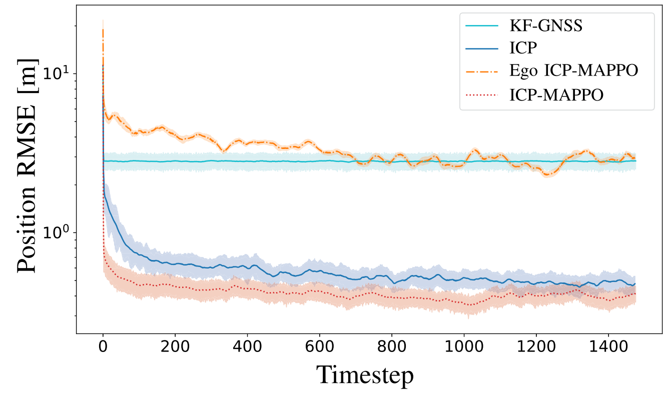
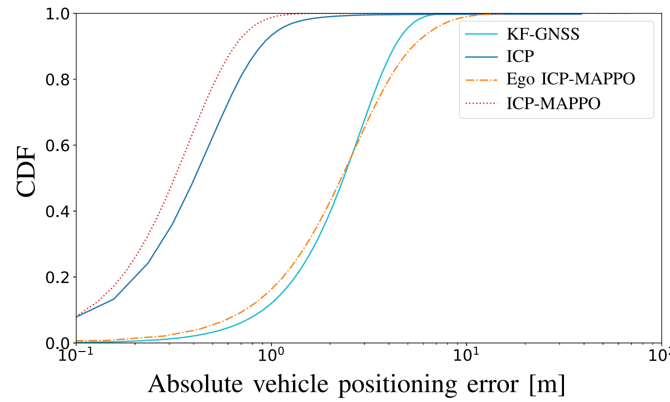


Fig. 7. Mean value function varying the number of training episodes.



(a)



(b)

Fig. 8. Testing performances on the cooperative scenario. (a) RMSE of the position over time for the single-agent KF-GNSS, ICP, proposed single agent and cooperative ICP-MAPPO. (b) CDF of the absolute error.

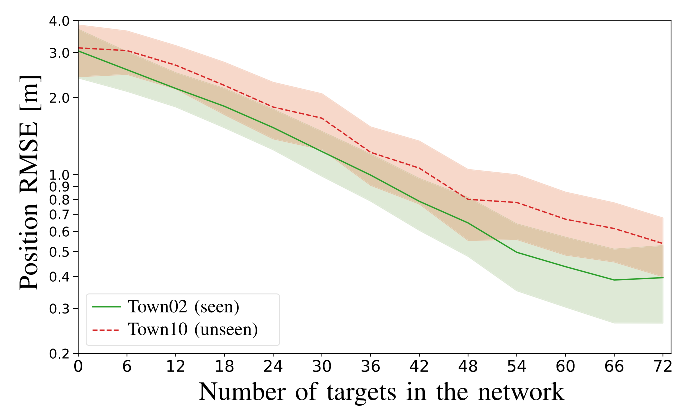


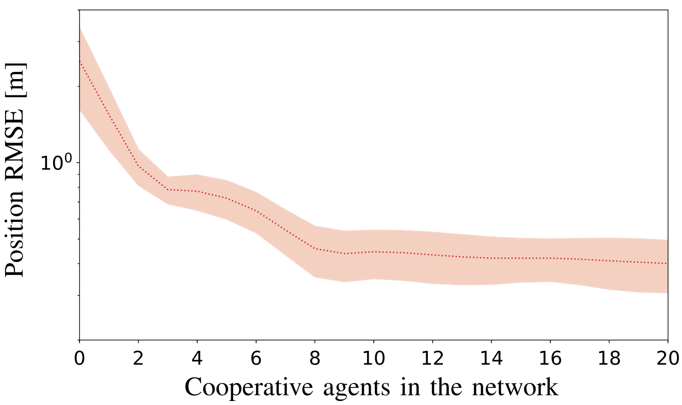
Fig. 9. RMSE on the position estimate achieved by ICP-MAPPO varying the number of targets (i.e., poles) in two distinct environments.

The metrics are computed among agents and trajectory over the whole episode. From the figures, we notice two distinct phases of the training: before and after reward convergence. In the first phase, i.e., before episode 250, the exploration is encouraged, leading to a much higher variability of the reward and a very rapid decrease of the LSTM loss function. After passing into the second phase, the positioning improvement becomes smaller, with a consequent convergence of the reward to the value of 2. Notably, also the mean value function converges after about 250 episodes, but with a high variance between agents and trajectories. This may be indicative of a rich and complex environment where the optimal policy may not be static, but rather dynamic and contingent on the interactions between agents and the environment. Indeed, the complexity of the state, e.g., each agent has a different trajectory in the space, can lead to a wide range of value function estimates as different states are visited with varying frequencies.

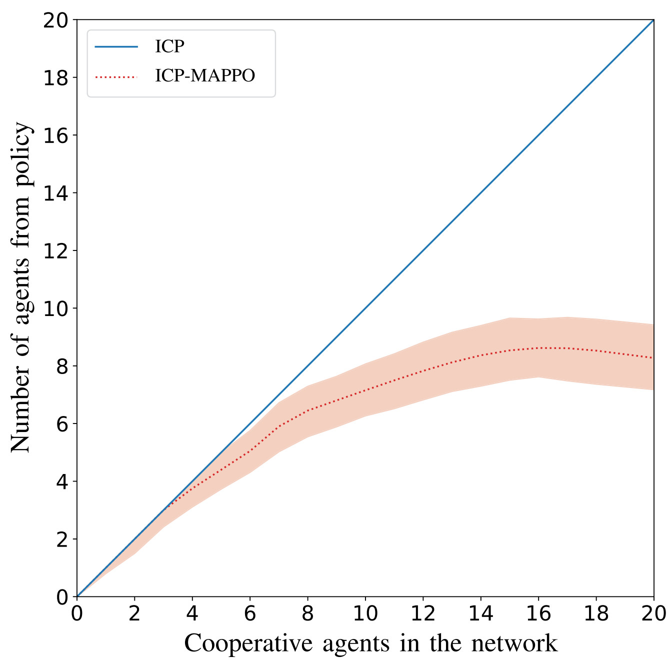
2) Cooperative Positioning Testing: This experiment has the objective of comparing the positioning capabilities of ICP-MAPPO with respect to the baselines in an unseen testing trajectory. To this aim, Fig. 8 shows the root mean square error (RMSE) on the vehicle position estimate at each timestep of the trajectory (Fig. 8(a)) and the corresponding cumulative density function (CDF) of the absolute error (Fig. 8(b)). The RMSE is computed among the agents at the single timestep, while the mean and error bounds are computed within the MC evaluations. From the results, we observe that the Ego ICP-MAPPO method, which only relies on GNSS measurements, converges to the KF-GNSS method, indicating a correct usage of the observations to estimate the position. Moving to the cooperative methods, we notice a higher speed of convergence of ICP-MAPPO with respect to the conventional ICP. This is mainly due to the learned vehicles' dynamics and to the effective combination of neighbors' observations. As a consequence, the ICP-MAPPO algorithm outperforms the ICP method in terms of absolute error by 21%, passing from a median of 42 cm to 33 cm.

3) Generalization Capabilities: This experiment aims at assessing the generalization capabilities of the proposed method in unseen scenarios. To evaluate the environmental dependence of our model, we tested the pre-trained ICP-MAPPO on a different CARLA map, specifically *Town10*. In Fig. 9, we plotted the position RMSE on testing trajectories in both *Town02* (used for training) and *Town10* (unseen environment), varying the number of passive objects in the respective map. We shall notice that the numbers of poles in *Town10* and *Town02* are 146 and 72, respectively. Since ICP-MAPPO was trained with a maximum input size of 72 measurements, we adjusted the number of targets up to 72 for this experiment.

The results in Fig. 9 confirm that, even in the unseen scenario, a higher number of vehicles increases the positioning accuracy thanks to the cooperation among vehicles. Comparing the results



(a)



(b)

Fig. 10. Communication efficiency comparison between the ICP and the proposed ICP-MAPPO methods. (a) RMSE on the position varying the maximum number of cooperative agents in the network. (b) Mean number of neighbor agents selected by the policy varying the maximum connectivity of the graph.

on *Town02* and *Town10*, we note that in the limit-case of no measurements shared among agents, the performances in the two scenarios coincide. On the contrary, when the number of features increases, the performances on the unseen scenario are slightly lower (i.e., about 10cm) despite the completely new environment.

4) Communication Efficiency: In this last assessment, we test the effectiveness of the policy choices in terms of cooperation power and communication efficiency. In Fig. 10 we report the position RMSE at convergence (Fig. 10(a)) and the mean number of selected agents from the policy (Fig. 10(b)) varying the maximum degree of connectivity allowed in the network. In Fig. 10(a) we observe an intuitive inverse relation between the maximum cooperative agents and the RMSE, with a rapid decrease under 1 m of RMSE with just 2 agents. Notably, after 8 cooperative agents, the improvement in RMSE is negligible, with convergence to about 40 cm. To study this behaviour, in

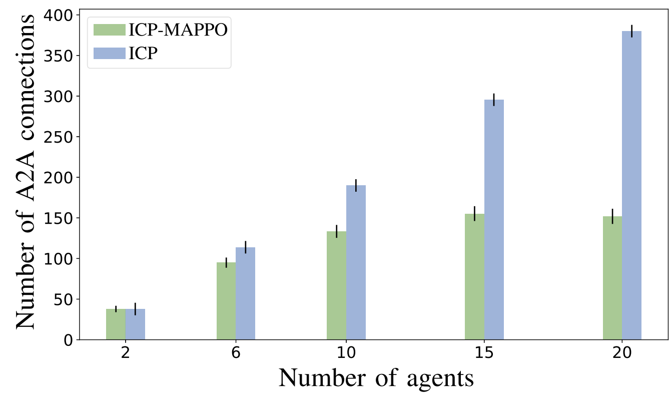


Fig. 11. Mean number of A2A connections in the network graph, for the ICP and the proposed ICP-MAPPO algorithms, and different maximum number of cooperative agents.

Fig. 10(b) we notice that the policy tends to select no more than 9 agents for cooperation. This likely occurs because the marginal benefits of additional cooperation diminish beyond this point, leading agents to prefer collaboration with only their closest neighbors. Indeed, incorporating data from distant agents that do not observe common targets results in only slight enhancements in positional accuracy. Lastly, we highlight that the ICP-MAPPO has higher performance than the ICP method for the same number of cooperative agents in the network.

To evaluate the trade-off between positioning accuracy and communication overhead, in Fig. 11, we plot the mean number of A2A links, considering varying numbers of cooperative vehicles in $\{2, 6, 10, 15, 20\}$. We observe that with a smaller number of cooperative agents, such as 2, the ICP-MAPPO tends to employ all available agents, leveraging neighbors' measurements to rapidly reduce GNSS uncertainty. Conversely, with a higher number of agents, particularly beyond 10, the benefits of additional cooperation decrease (as shown in Fig. 10(a)). This is because only the closest neighbors with a significant number of shared targets substantially enhance positioning accuracy. Notably, with 10 and 20 agents, ICP-MAPPO reduces the number of links by 30% and 60%, respectively, compared to ICP.

VI. Conclusion

In this paper, we addressed the problem of CP in a distributed network of agents that exploit passively detected targets as common reference points to improve the positioning accuracy according to the ICP framework. We provided a generalization of the Bayesian ICP solution by exploiting the MARL approach, which enables the dynamic optimization of the A2A links used for cooperation accounting for partial observability of the state. We presented a novel ICP-MAPPO algorithm where the agents actively select the neighbors to communicate with by following their optimized policy. This allows to minimize the communication overhead for cooperation, while improving the positioning accuracy of ego-agent systems. The proposed solution is proven to outperform single and multi-agent conventional approaches thanks to DL-based states' belief and policy models.

Realistic simulations of a C-ITS scenario created with CARLA simulator demonstrate the superior performances of ICP-MAPPO with state-of-the-art ICP methods, both in terms of positioning accuracy and efficiency of communications. The cooperation is indeed intelligently exploited to enhance the performances and, at the same time, the communication efficiency, by selecting ad-hoc neighbors that are relevant for the task. The benefits of the approach look promising for applications where groups of agents have a common inference objective and predictions/decisions need to be taken based on incomplete or uncertain data.

As future work, we envision the extension of the proposed method to decentralized frameworks [112], incorporating also data association of the targets to the measurements. Additionally, performances could be enhanced by exploiting a higher dimension of latent features within object detectors, instead of filtering specific objects such as poles. This approach would allow vehicles to exchange much more meaningful information in a compressed manner. Furthermore, including motion planning [113] could enable the system to not only estimate but also modify the vehicles' states according to their destinations. Finally, introducing safe RL [114] by adding safety constraints related to communication resources, such as maximum available bandwidth, would ensure that the policies learned by the agents remain efficient under real-world communication constraints.

Appendix A
Proof of (18)

To prove (18), we start by writing the gradient of the RL objective function in (17) as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) &= \nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})}\{\widetilde{R}(\boldsymbol{\tau})\} = \nabla_{\boldsymbol{\theta}} \sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\, \widetilde{R}(\boldsymbol{\tau}) \\ &= \sum_{\boldsymbol{\tau}} \nabla_{\boldsymbol{\theta}}\, p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\, \widetilde{R}(\boldsymbol{\tau}). \end{aligned} \tag{A1}$$

Now, we can rewrite the gradient of the trajectory PDF $\nabla_{\boldsymbol{\theta}}\, p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})$ using the log-derivative trick as

$$\nabla_{\boldsymbol{\theta}}\, p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}}) = p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} \log\left(p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\right). \tag{A2}$$

Given that the gradient of the log-trajectory PDF $\nabla_{\boldsymbol{\theta}} \log(p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}}))$ is

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \log\left(p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\right) &= \nabla_{\boldsymbol{\theta}} \log\left(T_0 \prod_{t=0}^{H-1} T(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)\, \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) \\ &= \sum_{t=0}^{H-1} \nabla_{\boldsymbol{\theta}} \log\left(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) \end{aligned} \tag{A3}$$

we can rewrite (A1) as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) &= \sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} \log\left(p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\right) \widetilde{R}(\boldsymbol{\tau}) \\ &= \mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})} \left\{\nabla_{\boldsymbol{\theta}} \log\left(p(\boldsymbol{\tau}|\pi_{\boldsymbol{\theta}})\right) \widetilde{R}(\boldsymbol{\tau})\right\} \\ &= \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim p(\boldsymbol{s}_t,\boldsymbol{a}_t|\pi_{\boldsymbol{\theta}})} \left\{ \sum_{t=0}^{H-1} \nabla_{\boldsymbol{\theta}} \log\left(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) \right. \\ &\qquad \left. \times \sum_{t=0}^{H-1} \gamma^t\, R(\mathbf{s}_t,\mathbf{a}_t) \right\}. \end{aligned} \tag{A4}$$

Since the action $\mathbf{a}_t$ at time t only influences the future rewards and not the past ones, (A4) can be equivalently rewritten as

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim p(\boldsymbol{s}_t,\boldsymbol{a}_t|\pi_{\boldsymbol{\theta}})} \left\{ \sum_{t=0}^{H-1} \nabla_{\boldsymbol{\theta}} \log\left(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) \mathsf{R}_t \right\} \tag{A5}$$

where we used the reward-to-go at time t $\mathsf{R}_t = \sum_{t'=t}^{H-1} \gamma^{t'-t}\, R(\mathbf{s}_{t'},\mathbf{a}_{t'})$, as opposed to R_0.

Since it can be proven that for any function of the state $B(\mathbf{s}_t)$ called baseline, we have that $\mathbb{E}_{\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \{\nabla_\theta \log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\, B(\mathbf{s}_t))\} = 0$, then we can reduce the variance of the PO algorithm, while remaining unbiased, by subtracting the baseline from the reward-to-go as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim p(\boldsymbol{s}_t,\boldsymbol{a}_t|\pi_{\boldsymbol{\theta}})} \left\{ \sum_{t=0}^{H-1} \Big[\nabla_{\boldsymbol{\theta}} \log\left(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) \right. \\ \left. \times \left(\mathsf{R}_t - B(\mathbf{s}_t)\right)\Big] \right\}. \end{aligned} \tag{A6}$$

Finally, R_t and $B(\mathbf{s}_t)$ are usually substituted with their estimates $Q^\pi(\mathbf{s}_t,\mathbf{a}_t)$ and $V^\pi(\mathbf{s}_t)$, respectively, leading to the definition of the advantage function $A_t = Q^\pi(\mathbf{s}_t,\mathbf{a}_t) - V^\pi(\mathbf{s}_t)$. Recently, more advanced versions of the advantage function, as the generalized advantage estimator (GAE) function A_t^{GAE} have been proposed in the literature [100] to regulate the bias-variance trade-off, increase stability, efficiency, and obtain faster convergence. We want to point out that usage of the baseline and/or the estimate of R_t are not necessary, and thus any function $F_t \in \{\mathsf{R}_t, Q^\pi(\mathbf{s}_t,\mathbf{a}_t), \mathsf{R}_t - V^\pi(\mathbf{s}_t), A_t, A_t^{\text{GAE}}\}$ is a valid choice.

References

[1] M. Z. Win et al., "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[2] L. Gao et al., "Cooperative localization in transportation 5.0," *IEEE Trans. Intell. Veh.*, vol. 9, no. 3, pp. 4259–4264, Mar. 2024.

[3] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1136–1165, Jul. 2018.

[4] Y. Gao, H. Jing, M. Dianati, C. M. Hancock, and X. Meng, "Performance analysis of robust cooperative positioning based on GPS/UWB integration for connected autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 790–802, Jan. 2023.

[5] A. Mahmoud, A. Noureldin, and H. S. Hassanein, "Integrated positioning for connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 397–409, Jan. 2020.

[6] 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.

[7] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, Jul. 2005.

[8] C. A. Gómez-Vega, Z. Liu, C. A. Gutiérrez, M. Z. Win, and A. Conti, "Efficient deployment strategies for network localization with assisting nodes," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 6272–6287, May 2024.

[9] L. Chen et al., “Milestones in autonomous driving and intelligent vehicles: Survey of surveys,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1046–1056, Feb. 2023.
[10] D. Cao et al., “Future directions of intelligent vehicles: Potentials, possibilities, and perspectives,” *IEEE Trans. Intell. Veh.*, vol. 7, no. 1, pp. 7–10, Mar. 2022.
[11] P. Yang, D. Duan, C. Chen, X. Cheng, and L. Yang, “Multi-sensor multi-vehicle (MSMV) localization and mobility tracking for autonomous driving,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14355–14364, Dec. 2020.
[12] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, “Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952–964, Feb. 2017.
[13] T. G. Reid et al., “Localization requirements for autonomous vehicles,” *SAE Int. J. Connected Automated Veh.*, vol. 2, no. 3, pp. 173–190, Oct. 2019.
[14] 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.
[15] M. H. C. Garcia et al., “A tutorial on 5G NR V2X communications,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1972–2026, Third Quarter 2021.
[16] F. Morselli, S. Modarres Razavi, M. Z. Win, and A. Conti, “Soft information-based localization for 5G networks and beyond,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9923–9938, Dec. 2023.
[17] H. Zhou, W. Xu, J. Chen, and W. Wang, “Evolutionary V2X technologies toward the internet of vehicles: Challenges and opportunities,” *Proc. IEEE*, vol. 108, no. 2, pp. 308–323, Feb. 2020.
[18] G. Torsoli, M. Z. Win, and A. Conti, “Blockage intelligence in complex environments for beyond 5G localization,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1688–1701, Jun. 2023.
[19] A. Conti, G. Torsoli, C. A. Gómez-Vega, A. Vaccari, G. Mazzini, and M. Z. Win, “3GPP-compliant datasets for xG location-aware networks,” *IEEE Open J. Veh. Technol.*, vol. 5, pp. 473–484, 2024.
[20] A. Conti, G. Torsoli, C. A. Gómez-Vega, A. Vaccari, and M. Z. Win, “XG-LOC: 3GPP-compliant datasets for xG location-aware networks,” *IEEE Dataport*, Dec. 2023, doi: 10.21227/rper-vc03.
[21] L. Italiano, B. C. Tedeschini, M. Brambilla, H. Huang, M. Nicoli, and H. Wymeersch, “A tutorial on 5G positioning,” *IEEE Commun. Surveys Tuts.*, pp. 1–48, Aug. 2024, doi: 10.1109/COMST.2024.3449031.
[22] B. Camajori Tedeschini, G. Kwon, M. Nicoli, and M. Z. Win, “Real-time bayesian neural networks for 6G cooperative positioning and tracking,” *IEEE J. Sel. Areas Commun.*, vol. 42, no. 9, pp. 2322–2338, Sep. 2024.
[23] A. Conti et al., “Location awareness in beyond 5G networks,” *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 22–27, Nov. 2021.
[24] B. C. Tedeschini and M. Nicoli, “Cooperative deep-learning positioning in mmWave 5G-advanced networks,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3799–3815, Dec. 2023.
[25] M. A. Javed, S. Zeadally, and E. B. Hamida, “Data analytics for cooperative intelligent transport systems,” *Veh. Commun.*, vol. 15, pp. 63–72, Jan. 2019.
[26] A. Alalewi, I. Dayoub, and S. Cherkaoui, “On 5G-V2X use cases and enabling technologies: A comprehensive survey,” *IEEE Access*, vol. 9, pp. 107710–107737, 2021.
[27] K. Sehla, T. M. T. Nguyen, G. Pujolle, and P. B. Velloso, “Resource allocation modes in C-V2X: From LTE-V2X to 5G-V2X,” *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8291–8314, Jun. 2022.
[28] M. Driusso, C. Marshall, M. Sabathy, F. Knutti, H. Mathis, and F. Babich, “Vehicular position tracking using LTE signals,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3376–3391, Apr. 2017.
[29] G. Kwon, Z. Liu, A. Conti, H. Park, and M. Z. Win, “Integrated localization and communication for efficient millimeter wave networks,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3925–3941, Dec. 2023.
[30] F. Liu et al., “Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.
[31] G. Kwon, A. Conti, H. Park, and M. Z. Win, “Joint communication and localization in millimeter wave networks,” *IEEE J. Sel. Top. Signal Process.*, vol. 15, no. 6, pp. 1439–1454, Nov. 2021.
[32] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs, and H. Wymeersch, “Implicit cooperative positioning in vehicular networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3964–3980, Dec. 2018.
[33] 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.
[34] 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, 2022.
[35] F. Jin, K. Liu, C. Liu, T. Cheng, H. Zhang, and V. C. S. Lee, “A cooperative vehicle localization and trajectory prediction framework based on belief propagation and transformer model,” *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2746–2758, Feb. 2024.
[36] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. Vincent Poor, “Network operation strategies for efficient localization and navigation,” *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, Jul. 2018.
[37] 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.
[38] W. Dai, Y. Shen, and M. Z. Win, “Distributed power allocation for cooperative wireless network localization,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 1, pp. 28–40, Jan. 2015.
[39] P. Yang, C. Xiang, and S. Zhang, “Distributed joint power and bandwidth allocation for multiagent cooperative localization,” *IEEE Commun. Lett.*, vol. 26, no. 11, pp. 2601–2605, Nov. 2022.
[40] T. Zhang, A. F. Molisch, Y. Shen, Q. Zhang, H. Feng, and M. Z. Win, “Joint power and bandwidth allocation in wireless cooperative localization networks,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6527–6540, Oct. 2016.
[41] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng, “A multimodality fusion deep neural network and safety test strategy for intelligent vehicles,” *IEEE Trans. Intell. Veh.*, vol. 6, no. 2, pp. 310–322, Jun. 2021.
[42] Z. Liu et al., “Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6640–6653, Jul. 2022.
[43] T. Wang, Y. Shen, A. Conti, and M. Z. Win, “Network navigation with scheduling: Error evolution,” *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7509–7534, Nov. 2017.
[44] B. Teague, Z. Liu, F. Meyer, A. Conti, and M. Z. Win, “Network localization and navigation with scalable inference and efficient operation,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2072–2087, Jun. 2022.
[45] S. Dwivedi, D. Zachariah, A. De Angelis, and P. Handel, “Cooperative decentralized localization using scheduled wireless transmissions,” *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1240–1243, Jun. 2013.
[46] L. Barbieri, B. Camajori Tedeschini, M. Brambilla, and M. Nicoli, “Deep learning-based cooperative LiDAR sensing for improved vehicle positioning,” *IEEE Trans. Signal Process.*, vol. 72, pp. 1666–1682, 2024.
[47] Y. Weiss and W. T. Freeman, “Correctness of belief propagation in gaussian graphical models of arbitrary topology,” *Neural Comput.*, vol. 13, no. 10, pp. 2173–2200, Oct. 2001.
[48] J. Yedidia, W. Freeman, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.
[49] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Found. Trends Mach. Learn.*, vol. 1, no. 1/2, pp. 1–305, Nov. 2008.
[50] F. Meyer et al., “Message passing algorithms for scalable multitarget tracking,” *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.
[51] G. Soldi, F. Meyer, P. Braca, and F. Hlawatsch, “Self-tuning algorithms for multisensor-multitarget tracking using belief propagation,” *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 3922–3937, Aug. 2019.
[52] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, “Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks,” *IEEE Access*, vol. 6, pp. 32328–32338, 2018.
[53] B. Camajori Tedeschini, M. Brambilla, L. Barbieri, G. Balducci, and M. Nicoli, “Cooperative lidar sensing for pedestrian detection: Data association based on message passing neural networks,” *IEEE Trans. Signal Process.*, vol. 71, pp. 3028–3042, 2023.
[54] B. Camajori Tedeschini, M. Brambilla, and M. Nicoli, “Message passing neural network versus message passing algorithm for cooperative positioning,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 6, pp. 1666–1676, Dec. 2023.

[55] B. Camajori Tedeschini, M. Brambilla, and M. Nicoli, “Split consensus federated learning: An approach for distributed training and inference,” *IEEE Access*, vol. 12, pp. 119535–119549, 2024.
[56] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction. A Bradford Book*. Cambridge, MA, USA: MIT Press, Oct. 2018.
[57] D. P. Bertsekas, *A Course in Reinforcement Learning*, 1st ed. Belmont, MA, USA: Athena Sci., Jun. 2023.
[58] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*, 1st ed. Belmont, MA, USA: Athena Sci., Jul. 2019.
[59] V. Mnih et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[60] D. P. Bertsekas, *Lessons from AlphaZero Optimal, Model Predictive, Adaptive Control*, 1st ed. Belmont, MA, USA: Athena Sci., Aug. 2021.
[61] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” in *Proc. 4th Int. Conf. Learn. Representations*, 2016, pp. 1–14.
[62] N. R. Ke et al., “Learning dynamics model in reinforcement learning by incorporating the long term future,” in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–14.
[63] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed, Belmont, MA, USA: Athena Sci., Oct. 2000.
[64] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volume II: Approximate Dynamic Programming*, 4th ed. Belmont, MA, USA: Athena Sci., Jun. 2012.
[65] N. C. Luong et al., “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, Fourth Quarter 2019.
[66] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. M. Leung, “Power control based on deep reinforcement learning for spectrum sharing,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4209–4219, Jun. 2020.
[67] F. Meng, P. Chen, L. Wu, and J. Cheng, “Power allocation in multi-user cellular networks: Deep reinforcement learning approaches,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6255–6267, Oct. 2020.
[68] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, and H. Li, “Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach,” *IEEE Access*, vol. 6, pp. 25463–25473, 2018.
[69] J. G. Vlachogiannis and N. D. Hatziargyriou, “Reinforcement learning for reactive power control,” *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1317–1325, Aug. 2004.
[70] L. Liang, H. Ye, and G. Y. Li, “Spectrum sharing in vehicular networks based on multi-agent reinforcement learning,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.
[71] O. Naparstek and K. Cohen, “Deep multi-user reinforcement learning for distributed dynamic spectrum access,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.
[72] H. Song, L. Liu, J. Ashdown, and Y. Yi, “A deep reinforcement learning framework for spectrum management in dynamic spectrum access,” *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11208–11218, Jul. 2021.
[73] Q. Luo, C. Li, T. H. Luan, and W. Shi, “Collaborative data scheduling for vehicular edge computing via deep reinforcement learning,” *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9637–9650, Oct. 2020.
[74] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, “Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications,” *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.
[75] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs, Ser. SpringerBriefs in Intelligent Systems*. Berlin, Germany: Springer, Jun. 2016.
[76] L. Kraemer and B. Banerjee, “Multi-agent reinforcement learning as a rehearsal for decentralized planning,” *Neurocomputing*, vol. 190, no. C., pp. 82–94, May 2016.
[77] S. Bhattacharya, S. Kailas, S. Badyal, S. Gil, and D. Bertsekas, “Multi-agent reinforcement learning: Rollout and policy iteration for POMDP with application to multirobot problems,” *IEEE Trans. Robot.*, vol. 40, pp. 2003–2023, 2024.
[78] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, “Deep decentralized multi-task multi-agent reinforcement learning under partial observability,” in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2681–2690.
[79] S. Gronauer and K. Diepold, “Multi-agent deep reinforcement learning: A survey,” *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 895–943, Apr. 2021.
[80] K. Zhang, Z. Yang, and T. Başar, *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*. Berlin, Germany: Springer, Jun. 2021, pp. 321–384.
[81] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Trans. Syst., Man, Cybern., Part C. (Appl. Rev.)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
[82] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, “Monotonic value function factorisation for deep multi-agent reinforcement learning,” *J. Mach. Learn. Res.*, vol. 21, no. 178, pp 1–51, 2020, Mar. 2020.
[83] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, “QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5887–5896.
[84] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, “QPLEX: Duplex dueling multi-agent q-learning,” in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 1–27.
[85] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Proc. 31th Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6382–6393.
[86] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, “Counterfactual multi-agent policy gradients,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2974–2982.
[87] C. Yu et al., “The surprising effectiveness of PPO in cooperative, multi-agent games,” in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 1–14.
[88] V. Mnih et al., “Asynchronous methods for deep reinforcement learning,” in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
[89] J. K. Gupta, M. Egorov, and M. Kochenderfer, “Cooperative multi-agent control using deep reinforcement learning,” in *Proc. Auton. Agents Multiagent Syst.*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds. Berlin, Germany, Nov. 2017, pp. 66–83.
[90] F. Christianos, G. Papoudakis, M. A. Rahman, and S. V. Albrecht, “Scaling multi-agent reinforcement learning with selective parameter sharing,” in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 1989–1998.
[91] J. K. Terry, N. Grammel, S. Son, and B. Black, “Parameter sharing for heterogeneous agents in multi-agent reinforcement learning, *ArXiv*, pp. 1–16, Oct. 2023.
[92] Z. Xia et al., “Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 931–945, Jan. 2022.
[93] B. Peng, G. Seco-Granados, E. Steinmetz, M. Frohle, and H. W. Wymeersch, “Decentralized scheduling for cooperative localization with deep reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4295–4305, May 2019.
[94] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proc. 1st Conf. Robot Learn.*, 2017, pp. 1–16.
[95] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[96] A. Tampuu et al., “Multiagent cooperation and competition with deep reinforcement learning,” *PLoS One*, vol. 12, Apr. 2017, Art. no. e0172395.
[97] N. Mehta, P. Tadepalli, and A. Fern, “Multi-agent shared hierarchy reinforcement learning,” in *Proc. ICML Workshop Rich Representations Reinforcement Learn.*, 2005, pp. 45–50.
[98] P. Sunehag et al., “Value-decomposition networks for cooperative multi-agent learning, 2017, *arXiv:1706.05296*.
[99] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Mach. Learn.*, vol. 8, no. 3-4, pp. 229–256, Jan. 2005.
[100] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” in *Proc. 4th Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1–14.
[101] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust region policy optimization,” in *Proc. 31st Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
[102] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms, 2017, *arXiv:1707.06347*.
[103] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine, “Model-based value estimation for efficient model-free reinforcement learning,” in *Proc. 35st Int. Conf. Mach. Learn.*, 2018, pp. 1–12.
[104] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, “Sample-efficient reinforcement learning with stochastic ensemble value expansion,” in *Proc. 31th Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8234–8244.
[105] D. P. Bertsekas, *Rollout, Policy Iteration, and Distributed Reinforcement Learning*, 1st ed. Belmont, MA, USA: Athena Sci., Aug. 2020.

[106] A. Ilyas et al., “A closer look at deep policy gradients,” in *Proc. 8th Int. Conf. Learn. Representations*, 2020, pp. 1–27.

[107] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[108] J. H. Jhung Suk, H. Park, and S. Kim, “Hardware accelerators for autonomous vehicles,” in *Artificial Intelligence and Hardware Accelerators*, A. J. Mishra Cha, H. Park, and S. Kim Eds. Cham, Switzerland: Springer, 2023, pp. 269–317.

[109] M. Mikami, K. Serizawa, Y. Ishida, H. Nishiyori, K. Moto, and H. Yoshino, “Field experimental evaluation on latency and reliability performance of 5G NR V2V direct communication in real express highway environment,” in *Proc. IEEE 91st Veh. Technol. Conf.*, May 2020, pp. 1–5.

[110] B. Coll-Perales et al., “End-to-end V2X latency modeling and analysis in 5G networks,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5094–5109, Apr. 2023.

[111] 3GPP, “Study on enhancement of 3GPP Support for 5G V2X Services,” 3rd Gen. Partnership Project, Sophia Antipolis, France, Tech. Rep. 22.886 Version 16.2.0, 2018 .

[112] D. Chen, K. Zhang, Y. Wang, X. Yin, Z. Li, and D. Filev, “Communication-efficient decentralized multi-agent reinforcement learning for cooperative adaptive cruise control,” *IEEE Trans. Intell. Veh.*, vol. 9, no. 10, pp. 6436–6449, Oct. 2024.

[113] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, and Y. Zhao, “Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5435–5444, Dec. 2021.

[114] M. Han, Y. Tian, L. Zhang, J. Wang, and W. Pan, “Reinforcement learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee,” *Automatica*, vol. 129, May 2021, Art. no. 109689.

Bernardo Camajori Tedeschini (Graduate Student Member, is pursuing the Ph.D. degree in Information Technology at the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milan, Italy, since November 2021. He received his M.Sc. (Hons.) degree in Telecommunications Engineering and B.Sc. (Hons.) degree in Computer Science from the Politecnico di Milano, Milan, Italy, in 2021 and 2019, respectively.

He is currently a visiting student with the Wireless Information and Network Sciences Laboratory at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. In 2021, he has served as a Visiting Research Scientist at CERN, Geneva, Switzerland, where he worked on the CAFEIN project, focusing on the development and deployment of a Federated network platform. His research interests encompass federated learning, machine learning for signal processing and sensing over networks, and localization methods.

Mr. Camajori Tedeschini is a recipient of a Ph.D. grant from Italy’s Ministero dell’Istruzione, dell’Università e della Ricerca (MIUR) and the Roberto Rocca Doctoral Fellowship, which was jointly awarded by MIT and Politecnico di Milano. He earned both his Bachelor’s and Master’s degrees with highest honors and he was honored with the best freshmen prize from Politecnico di Milano in 2017.

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 the faculty of Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB) at the Politecnico di Milano as Research Fellow. His research interests include signal processing, statistical learning, and data fusion for cooperative localization and communication

Dr. Brambilla was the recipient of the Best Student Paper Award at the 2018 IEEE Statistical Signal Processing Workshop.

Monica Nicoli (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in communication engineering from 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 Politecnico di Milano as a Faculty Member. 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). She was a recipient of the Marisa Bellisario Award, in 1999, and a co-recipient of the best paper awards of the EuMA Mediterranean Microwave Symposium, in 2022, 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.

Moe Z. Win (Fellow, IEEE) is the Robert R. Taylor Professor at the Massachusetts Institute of Technology (MIT) and the founding director of the Wireless Information and Network Sciences Laboratory. Prior to joining MIT, he was with AT&T Research Laboratories and with the NASA Jet Propulsion Laboratory.

His research encompasses fundamental theories, algorithm design, and network experimentation for a broad range of real-world problems. His current research topics include ultra-wideband systems, network localization and navigation, network interference exploitation, and quantum information science. He has served the IEEE Communications Society as an elected Member-at-Large on the Board of Governors, as elected Chair of the Radio Communications Committee, and as an IEEE Distinguished Lecturer. Over the last two decades, he held various editorial positions for IEEE journals and organized numerous international conferences. He has served on the SIAM Diversity Advisory Committee.

Dr. Win is an elected Fellow of the AAAS, the EURASIP, the IEEE, and the IET. He was honored with two IEEE Technical Field Awards: the IEEE Kiyo Tomiyasu Award (2011) and the IEEE Eric E. Sumner Award (2006, jointly with R. A. Scholtz). His publications, coauthored with students and colleagues, have received several awards. Other recognitions include the MIT Frank E. Perkins Award (2024), the MIT Everett Moore Baker Award (2022), the IEEE Vehicular Technology Society James Evans Avant Garde Award (2022), the IEEE Communications Society Edwin H. Armstrong Achievement Award (2016), the Cristoforo Colombo International Prize for Communications (2013), the Copernicus Fellowship (2011) and the *Laurea Honoris Causa* (2008) from the Università degli Studi di Ferrara, and the U.S. Presidential Early Career Award for Scientists and Engineers (2004).