Cooperative Positioning with Multi-Agent Reinforcement Learning

Bernardo Camajori Tedeschini*†, Mattia Brambilla*, Monica Nicoli‡, Moe Z. Win†
*Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy
†Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA
‡Dipartimento di Ingegneria Gestionale (DIG), Politecnico di Milano, 20133 Milan, Italy
Email: bernardo.camajori@polimi.it, mattia.brambilla@polimi.it, monica.nicoli@polimi.it, moewin@mit.edu

***Abstract*—In recent years, cooperative positioning technologies have emerged as promising augmentation systems for providing high-accuracy positioning (HAP) in cooperative intelligent transportation systems (C-ITS). Among the approaches, implicit cooperative positioning (ICP) takes advantage of shared target detections between vehicles to create common reference points for localization refinement. Their performance, however, is limited by reliance on predefined parametric models, low scalability and communication overhead. To address these problems, this paper introduces a deep multi-agent reinforcement learning (MARL) framework modelled as a decentralized-partially observable Markov decision process (Dec-POMDP). We propose an ICP-multi-agent proximal policy optimization (MAPPO) algorithm, where distributed agents (i.e., the connected vehicles) learn their dynamics and those of the surrounding targets by performing belief estimation over dynamic cooperation graphs that are continuously adjusted by de/activating communication links with neighbors agents. A C-ITS scenario is simulated in a CARLA environment accounting for realistic vehicle dynamics and inter-vehicle communications. The findings reveal that our ICP-MAPPO algorithm, leveraging dynamic decentralized execution and centralized training, outperforms ICP in terms of positioning accuracy and communication efficiency.**

***Index Terms*—Autonomous agents, cooperative positioning, multi-agent reinforcement learning, CARLA simulator.**

I. Introduction

In an era where automated mobility services are increasingly prevalent, the accuracy of localization technologies has become of paramount importance [1]. New-generation connected vehicles integrate advanced sensor suites and 5th generation (5G) vehicle-to-everything (V2X) communications to enhance connectivity and positioning services, setting a new standard for mobile localization in cooperative intelligent transport systemss (C-ITSs) [2]–[4]. Cooperative positioning (CP) emerges as a promising approach to enhance localization accuracy through inter-vehicle sidelink communications, sharing information about passive objects to mitigate global navigation satellite systems (GNSS) degradation [5]–[7].

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, in part by the project Centro Nazionale per la Mobilità Sostenibile (MOST), spokes 6 and 9, funded by the Italian Ministry of University and Research under the PNRR funding program, in part by the National Science Foundation under Grant CNS-2148251, and in part by the Federal Agency and Industry Partners in the RINGS Program.

Traditional CP methods leverage passive objects cooperatively detected in the surroundings as anchor points to be used for refining vehicle positioning by Bayesian-filtering. Framework of this type are the cooperative simultaneous localization and mapping (SLAM) [8], [9] and the implicit cooperative positioning (ICP) [10]–[12]. ICP methodologies have shown to outperform cooperative SLAM in terms of efficiency and accuracy in vehicle positioning [13]. However, these methods strive with high computational complexity and scalability issues when aggregating data across multiple vehicles. Despite attempts to address these limitations through distributed message passing algorithm (MPA) [14] or dynamic model adjustments [15], challenges in communication overhead and computational burden persist, hindering scalability.

The integration of machine learning (ML) and, more specifically, multi-agent reinforcement learning (MARL) into CP represents a paradigm shift in addressing the limitations of traditional Bayesian methods. ML provides innovative solutions for handling scalability in complex graphs and non-linearity/Gaussianity in models [16]–[18], with MARL and related deep learning (DL) variants being particularly effective in complex decision-making environments where independent agents share a common objective (i.e., reward) and decisions (i.e., actions) are based on incomplete or uncertain data about the system state [19]. In the literature of CP, such framework, namely decentralized-partially observable Markov decision process (Dec-POMDP) [20], has been applied either to intelligent swarm-based systems [21], such as unmanned aerial vehicles (UAVs), for tracking purposes, or for enhancing conventional Bayesian filtering through agent scheduling strategies [22], but not concurrently. On the contrary, MARL algorithms specialized on communications completely discard the state estimation part [23] and do not focus on communication efficiency but rather on how to optimally weigh the received information from the neighbors [24].

This paper aims at combining the two aspects, i.e., the estimation of the agents' state and the optimization of the agent-to-agent cooperation graph in a unique dynamic MARL framework In particular, we present a novel MARL algorithm, referred to as ICP-multi-agent proximal policy optimization (MAPPO), designed specifically for efficient distributed and cooperative position estimation. We formulate agent-specific

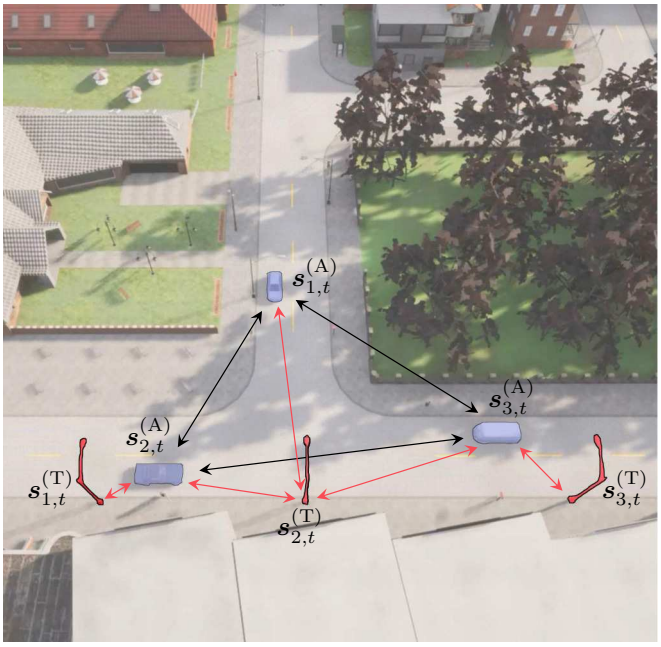


Fig. 1. 3D representation of the scenario with three vehicles and three objects (snapshot extracted from CARLA). A2A communication links and A2T detections are indicated with black and red arrows, respectively.

policies for optimizing communication scheduling among neighboring agents while concurrently acquiring an understanding of the environmental dynamics through the integration of neighbors' observations. We adopt a centralized-training to learn beliefs based on minimum mean square error (MMSE) criterion and policies from MAPPO objective functions. Then, during dynamic-decentralized-execution, the network graph is modified according to the agents actions and the state is estimated from the beliefs. Through rigorous validation in realistic C-ITS scenarios simulated with Carla [25] (see Fig. 1), our approach demonstrates superior performance in positioning accuracy and communication efficiency compared to existing ICP state-of-the-art solutions.

The structure of the article is as follows: Sec. II outlines the system model involving cooperative agents. Sec. III presents the the MARL framework and the proposed ICP-MAPPO algorithm for CP. Sec. IV delineates the simulation setup and discusses the outcomes of the experiments. Lastly, Sec. V draws the conclusions.

Notations: 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. The function $p_{\mathsf{x}}(x)$, and simply $p(x)$ when there is no ambiguity, denotes the probability density function (PDF) of x. 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 random variable, respectively. $\mathbb{R}$ stands for the set of real numbers.

II. System model

We indicate the vehicular network graph at time t as $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where $\mathcal{V} = \{1, \dots, N\}$ denotes the set of vehicles or agents (i.e., the nodes), and $\mathcal{E}_t$ represents the communication links among them (i.e., the edges). Each agent $i \in \mathcal{V}$ has a set $\mathcal{N}_{i,t}$ of neighboring agents at time t and it is described by a state $\mathbf{s}^{(\mathrm{A})}_{i,t} \in \mathbb{R}^{4\times 1}$, comprising the 2D position and velocity vectors, respectively, within a global coordinate framework. The collective state of all vehicles at time t is denoted as $\mathbf{s}^{(\mathrm{A})}_t = \left[\mathbf{s}^{(\mathrm{A})}_{i,t}\right]_{i=1}^{N}$. The kinematic state transition for vehicle i at time t follows the model:

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

with $\mathbf{w}^{(\mathrm{A})}_{i,t-1}$ encapsulating the driving noise, which accounts for motion uncertainty. The state-transition PDF following from the above model is $T\left(\boldsymbol{s}^{(\mathrm{A})}_{i,t} | \boldsymbol{s}^{(\mathrm{A})}_{i,t-1}\right) \triangleq p\left(\boldsymbol{s}^{(\mathrm{A})}_{i,t} | \boldsymbol{s}^{(\mathrm{A})}_{i,t-1}\right)$.

The environment also includes K static and passive objects (targets), gathered in the set $\mathcal{F} = \{1, \dots, K\}$, which vehicles can detect. In this work, we select roadside poles as detectable objects as they are widely spread and easily recognizable, especially in urban mobility contexts. Generalization to other kind of objects is possible for different scenarios. Each pole k has a constant 2D position state $\mathbf{s}^{(\mathrm{T})}_{k,t} \in \mathbb{R}^{2\times 1}$, and $\mathbf{s}^{(\mathrm{T})}_t = \left[\mathbf{s}^{(\mathrm{T})}_{k,t}\right]_{k\in\mathcal{F}}$ aggregates all passive object states at time t. Lastly, the aggregated state of the system is $\mathbf{s}_t = \left[\mathbf{s}^{(\mathrm{A})\top}_t \mathbf{s}^{(\mathrm{T})\top}_t\right]^\top$.

Vehicles are equipped with a GNSS receiver, a proximity positioning system, and a passive sensing technology such as radio detection and ranging (RADAR), light detection and ranging (LIDAR) or camera. The GNSS receiver provides a vehicle's state estimate $\mathbf{s}^{(\mathrm{A})}_{i,t}$, modeled as:

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

with $\mathbf{n}^{(\mathrm{GNSS})}_{i,t} \sim \mathcal{N}\left(\mathbf{0}_{2\times 2}, \boldsymbol{R}^{(\mathrm{GNSS})}_{i,t}\right)$ representing zero-mean Gaussian noise and and $\boldsymbol{H} = [\boldsymbol{I}_2\, \mathbf{0}_{2\times 2}] \in \mathbb{R}^{2\times 4}$. The proximity positioning system permits agent-to-agent (A2A) relative position measurement between vehicles $(i, j) \in \mathcal{E}_t$:

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

being $\mathbf{n}^{(\mathrm{A2A})}_{i,j,t} \sim \mathcal{N}\left(\mathbf{0}_{2\times 2}, \boldsymbol{R}^{(\mathrm{A2A})}_{i,j,t}\right)$ an additive zero-mean Gaussian noise term. The passive sensor enables agent-to-target (A2T) measurements for detecting passive objects $k \in \mathcal{F}_{i,t}$ within vehicle proximity, formulated as:

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

where $\mathbf{n}^{(\mathrm{A2T})}_{i,k,t} \sim \mathcal{N}\left(\mathbf{0}_{2\times 2}, \boldsymbol{R}^{(\mathrm{A2T})}_{i,k,t}\right)$ represents zero-mean Gaussian noise. From (2), (3) and (4), we defined their corresponding likelihoods $p\left(\boldsymbol{o}^{(\mathrm{GNSS})}_{i,t} | \boldsymbol{s}^{(\mathrm{A})}_{i,t}\right)$, $p\left(\boldsymbol{o}^{(\mathrm{A2A})}_{i,j,t} | \boldsymbol{s}^{(\mathrm{A})}_{i,t}, \boldsymbol{s}^{(\mathrm{A})}_{j,t}\right)$ and $p\left(\boldsymbol{o}^{(\mathrm{A2T})}_{i,k,t} | \boldsymbol{s}^{(\mathrm{A})}_{i,t}, \boldsymbol{s}^{(\mathrm{T})}_{k,t}\right)$, respectively. The vector of all available measurements of vehicle i at time t is denoted as $\mathbf{o}_{i,t} = \left[\mathbf{o}^{(\mathrm{GNSS})\top}_{i,t} \mathbf{o}^{(\mathrm{A2A})\top}_{i,t} \mathbf{o}^{(\mathrm{A2T})\top}_{i,t}\right]^\top$, with $\mathbf{o}^{(\mathrm{A2A})}_{i,t} = \left[\mathbf{o}^{(\mathrm{A2A})}_{i,j,t}\right]_{j\in\mathcal{N}_{i,t}}$ and $\mathbf{o}^{(\mathrm{A2T})}_{i,t} = \left[\mathbf{o}^{(\mathrm{A2T})}_{i,k,t}\right]_{k\in\mathcal{F}_{i,t}}$.

Given the likelihood of all measurements $O(\boldsymbol{o}_t|\boldsymbol{s}_t) \triangleq p(\boldsymbol{o}_t|\boldsymbol{s}_t)$ and the state transition PDF $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$, the objective of CP is to estimate $\widehat{\boldsymbol{s}}_t$ according to the MMSE criterion as:

$$\widehat{\boldsymbol{s}}_t = \mathbb{E}\{\mathbf{s}_t|\mathbf{o}_{1:t}\} = \int \boldsymbol{s}_t \, p(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}) \, \mathrm{d}\boldsymbol{s}_t \tag{5}$$

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

$$p(\boldsymbol{s}_t|\boldsymbol{o}_{1:t}) \propto p(\boldsymbol{o}_t|\boldsymbol{s}_t) \int p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) \, p(\boldsymbol{s}_{t-1}|\boldsymbol{o}_{1:t-1}) \, \mathrm{d}\boldsymbol{s}_{t-1} \, . \tag{6}$$

We denote the marginal posterior PDF of agent i (or *belief*) with $b(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{1:t}) \triangleq p(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{1:t})$. Note that this model assumes vehicles can accurately associate A2T measurements to known targets, focusing on CP challenges without the complexities of data association. Conventional ICP solutions solve this problem by performing either a centralized extended Kalman filter (EKF) [12] or a distributed MPA up to convergence [10]. In both cases, the solution is optimal only in the presence of linear and Gaussian models. This paper focuses on developing a MARL algorithm that enables agents to automatically learn the underlying models in non-linear and non-Gaussian settings and to improve communication efficiency through selective interaction with neighbors.

III. MARL for Methodology

A. MARL Framework

The framework for cooperative multi-agent systems (MAS) is conceptualized as a finite-horizon Dec-POMDP, represented by the tuple $\langle \mathcal{V}, \mathcal{S}, \mathcal{A}, T_0, T, \mathcal{O}, O, R, \gamma, H \rangle$. Here, $\mathcal{V}$ encapsulates the cooperative agents (i.e., vehicles), while $\mathcal{S}$ and $\mathcal{A}$ indicate the state and action spaces, respectively. T_0 specifies the initial state distribution at $t = 0$, and $T(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_t) = p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_t)$ describes the state transition PDF, integrating the actions $\boldsymbol{a}_t = \left[\boldsymbol{a}_{i,t}\right], i \in \mathcal{V}$, into the state dynamics $\boldsymbol{s}_t$. Each timestep t brings a joint observation $\boldsymbol{o}_t \in \mathcal{O}$, derived from $O(\boldsymbol{o}_t|\boldsymbol{a}_{t-1}, \boldsymbol{s}_t) = p(\boldsymbol{o}_t|\boldsymbol{a}_{t-1}, \boldsymbol{s}_t)$, and a reward r_t from a reward function $R(\boldsymbol{s}_t, \boldsymbol{a}_t) = r_t$. Lastly, γ and H are the discount factor and time horizon of the episode, respectively. Here, with episode, we indicate the duration of the simulation or real-world application within which agents operate, reflecting the predetermined number of timesteps over which strategic decisions and actions are executed.

Given the partial observability of states and rewards, agents maintain *histories* $\boldsymbol{h}_{i,1:t} = \boldsymbol{h}_{i,t} = \left[(\boldsymbol{a}_{i,t'-1}, \boldsymbol{o}_{i,t'})\right]_{t'=1}^{t}$, encapsulating past actions and observations. State estimates $\widehat{\boldsymbol{s}}_{i,t}$ are inferred using MMSE from the belief $b_{\boldsymbol{\psi}}(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{i,t}, \boldsymbol{a}_{i,t-1}, \boldsymbol{h}_{i,t-1})$ parameterized by $\boldsymbol{\psi}$, with actions sampled from the policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}_{i,t})$ parameterized by $\boldsymbol{\theta}$. The objective of the MARL method is to maximize the expected cumulative discounted reward as $\max_\pi J(\pi) = \mathbb{E}\{\mathsf{R}_0\}$, where $\mathsf{R}_t = \sum_{t'=t}^{H-1} \gamma^{t'-t} \, \mathsf{r}_{t'}$, called *reward-to-go*, is the cumulative discounted reward from time t to the end of the episode.

B. ICP-MAPPO

For solving the CP problem, while optimizing the communication/cooperation graph among agents, we propose the following Dec-POMDP:

1) Agents: Each agent in this model corresponds to a vehicle, indexed by $i \in \mathcal{V}$, participating in the network.

2) Actions: The action taken by agent i at time t, namely $\mathbf{a}_{i,t} = \left[\mathsf{a}_{i,j,t}\right]_{j=1}^{N}$, involves a binary decision $\mathsf{a}_{i,j,t} \in \{0, 1\}$ on whether to establish communication with agent j. This permits to actively modify the connectivity graph and thus enabling a better communication efficiency.

3) States: The states are $\mathbf{s}_t^{(\mathrm{A})}$, while the states of targets $\mathbf{s}_t^{(\mathrm{T})}$ are implicitly learned by the latent features of the neural networks (NNs).

4) Observations: Observations available to each vehicle, denoted as $\mathbf{o}_{i,t}$, encompass GNSS, A2A, and A2T measurements.

The general scheme for the Dec-POMDP is shown in Fig. 2. Note that neither the state transition of the environment, nor the rewards are observed by the agents. On the contrary, only observations are gathered and stored by the agents within histories, which are then used for action and state estimation. Therefore, as in the state-of-the-art literature of MARL, for ICP-MAPPO, we adopt a centralized-training procedure, enabling the agents to perform policy optimization and belief optimization, while having access to the full observable state $\mathbf{s}_t$ and measurements $\mathbf{o}_t$. Afterwords, the learned policies are deployed independently among agents, exploiting the modification of the coordination graph's structure based on the agents' actions, ranging from a fully-connected to a fully-decentralized configuration. Therefore, we call this approach *centralized-training and dynamic-decentralized-execution*.

The ICP-MAPPO for execution is composed of a long short-term memory (LSTM) and multi-layer perceptron (MLP) models for belief and action predictions, respectively, as:

$$\widehat{\boldsymbol{s}}_{i,t}, \boldsymbol{h}_{i,t}^{\mathrm{b}} = b_{\boldsymbol{\psi}}(\boldsymbol{s}_{i,t}|\boldsymbol{o}_{i,t}, \bar{\boldsymbol{a}}_{i,t-1}, \bar{\boldsymbol{h}}_{i,t-1}^{\mathrm{b}}) \tag{7}$$

$$\boldsymbol{a}_{i,t} \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_{i,t}|\boldsymbol{h}_{i,t}^{\mathrm{b}}) \, . \tag{8}$$

$\bar{\boldsymbol{a}}_{i,t} = \left[\bar{a}_{i,j,t}\right]_{j=1}^{N}$ denotes the adjusted action set for agent i at time t and it is derived by sampling actions from the policy distribution while also considering the network's connectivity constraints:

$$\bar{a}_{i,j,t} = \begin{cases} a_{i,j,t} & \text{if} \quad j \in \mathcal{N}_{i,t} \, , \\ -1 & \text{otherwise} \, . \end{cases} \tag{9}$$

Furthermore, $\bar{\boldsymbol{h}}_{i,t}^{\mathrm{b}}$ encapsulates the belief LSTM's hidden features, offering a condensed representation of the interaction histories between agent i and its chosen neighbors up to the previous timestep:

$$\bar{\boldsymbol{h}}_{i,t}^{\mathrm{b}} = \frac{\boldsymbol{h}_{i,t}^{\mathrm{b}} + \sum_{j \in \mathcal{V}} \boldsymbol{h}_{j,t}^{\mathrm{b}} \, \mathbf{1}(\bar{a}_{i,j,t} == 1)}{1 + \sum_{j \in \mathcal{V}} \mathbf{1}(\bar{a}_{i,j,t} == 1)} \tag{10}$$

where $\mathbf{1}(\cdot)$ acts as the indicator function, yielding 1 if the specified condition is met and 0 otherwise. We note that the

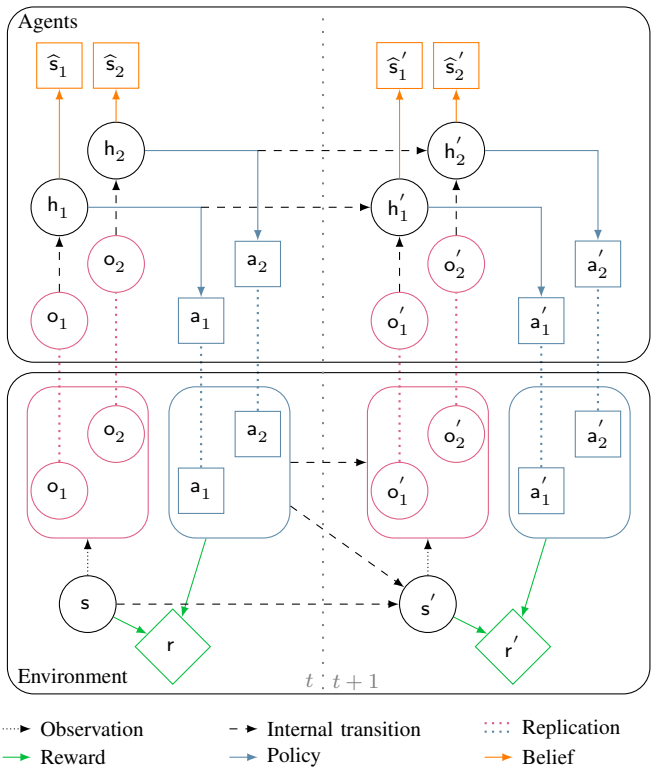


Fig. 2. Dec-POMDP scheme adopted in the ICP-MAPPO algorithm for agent and environment evolutions. Superscript $(\cdot)'$ stands for $t+1$ for graphical purposes.

action decisions at time t in (8) rely predominantly on information from the previous timestep, $\bar{\boldsymbol{h}}^{\mathrm{b}}_{i,t-1}$. This is because agent i cannot preemptively access its neighbors' measurements, $\boldsymbol{h}^{\mathrm{b}}_{j,t}, \forall j \in \mathcal{V}$, to decide on communication actions. The actions $\bar{\boldsymbol{a}}_{i,t}$ play two essential roles within the belief LSTM framework. First, they enable to incorporate which agents have been chosen for measurement fusion, crucial for making accurate state predictions. Second, by assigning negative values to actions when connectivity is not possible, each agent can implicitly discern its identification or index, enabling scalable and efficient training with parameter sharing [26].

In selecting the MARL algorithm, we favored policy optimization (PO) methods over Q-learning due to the latter's inherent bias issues when integrated with DL, leading to inaccuracies in estimating state-action values or Q-values. In contrast, PO algorithms exhibit significantly lower bias by directly optimizing the objective function $J(\pi)$ and have demonstrated superior performance in MARL contexts [27]. Although PO algorithms are characterized by high variance, necessitating extensive samples for convergence, this challenge is addressable through the learning of value functions, such as $V^{\pi}(\boldsymbol{s}_t)$ or $Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, which predict the expected long-term rewards for given states or state-action pairs. Specifically, we employ MAPPO [27] value estimation modelled with a recurrent neural network (RNN) with parameters $\boldsymbol{\phi}$ as:

$$\widehat{V}_{\boldsymbol{\phi}}(\boldsymbol{s}_{i,t}, \boldsymbol{h}^{\mathrm{V}}_{i,t-1}), \boldsymbol{h}^{\mathrm{V}}_{i,t} = V_{\boldsymbol{\phi}}(\boldsymbol{s}_{i,t}, \boldsymbol{h}^{\mathrm{V}}_{i,t-1}) \tag{11}$$

where $\boldsymbol{h}^{\mathrm{V}}_{i,t}$ are the hidden features of $V_{\boldsymbol{\phi}}$, also called *critic*. On the contrary, $\pi_{\boldsymbol{\theta}}$ is referred to as *actor*.

The actor and the critic are optimized according to the loss functions in MAPPO, with the difference that here the actor is a MLP since the hidden histories in the execution are contained in the belief LSTM. We define the reward function to incentive actions that lead to a specified improvement β in positioning accuracy at future timesteps. Essentially, each agent i evaluates whether choosing a different agent j' over agent j would have resulted in better performance. This concept is encapsulated in the following reward structure:

$$r_t = \begin{cases} -1 & \text{if} \quad \left\|\boldsymbol{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2 - \left\|\boldsymbol{s}_{t+1} - \widehat{\boldsymbol{s}}_{t+1}\right\|_2^2 \leq -\beta \\ +1 & \text{if} \quad \left\|\boldsymbol{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2 - \left\|\boldsymbol{s}_{t+1} - \widehat{\boldsymbol{s}}_{t+1}\right\|_2^2 > \beta \\ +2 & \text{if} \quad -\beta < \left\|\boldsymbol{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2 - \left\|\boldsymbol{s}_{t+1} - \widehat{\boldsymbol{s}}_{t+1}\right\|_2^2 \leq \beta. \end{cases} \tag{12}$$

Initially, a reward of -1 is assigned if the action deteriorates positioning accuracy by more than β. A reward of $+1$ is granted for an improvement exceeding β, and as the learning progresses towards convergence with smaller improvements, a long-term reward of $+2$ is introduced to encourage sustained accuracy gains. The belief function $b_{\boldsymbol{\psi}}$ utilizes a mean square error (MSE) loss function as:

$$L(\boldsymbol{\psi}) = \frac{1}{N L_\tau} \sum_{i \in \mathcal{V}} \sum_{\ell=1}^{L_\tau} \left\|\widehat{\boldsymbol{s}}_{i,t} - \boldsymbol{s}_{i,t}\right\|_2^2 \tag{13}$$

where L_τ is the length of a trajectory.

The pseudo-code for the ICP-MAPPO is reported in Algorithm 1, where $\boldsymbol{\tau}_t$ is a transitions defined as $\boldsymbol{\tau}_t = (\boldsymbol{s}_t, \boldsymbol{o}_t, \boldsymbol{h}^{\mathrm{b}}_t, \bar{\boldsymbol{h}}^{\mathrm{b}}_t, \boldsymbol{h}^{\mathrm{V}}_t, \boldsymbol{a}_t, \bar{\boldsymbol{a}}_t, r_t, \boldsymbol{s}_{t+1}, \boldsymbol{o}_{t+1}, \widehat{\boldsymbol{s}}_{t+1})$ and $\widehat{A}_{i,\ell} = R_\ell - \widehat{V}_{\boldsymbol{\phi}_{\mathrm{old}}}(\boldsymbol{s}_{i,\ell}, \boldsymbol{h}^{\mathrm{V}}_{i,\ell-1})$ is the advantage function estimate. ICP-MAPPO algorithm is an on-policy, low-bias algorithm, leveraging latest policy-generated data for agent training. A centralized value function, incorporating the state $\boldsymbol{s}_{i,t}$ in (11) beyond local observations, aids in precise value estimation. The belief computation aligns with model-based value estimation (MBVE) in reinforcement learning (RL) [28], utilizing learned dynamics for state prediction to reduce variance without introducing bias. Finally, rewards in (12) are linked to belief improvements in future timesteps rather than direct action outcomes, permitting to optimize actions for the CP objective function $\min_b J(b) = \min_b \mathbb{E}\left\{\sum_t \left\|\mathbf{s}_t - \widehat{\boldsymbol{s}}_t\right\|_2^2\right\}$.

IV. Simulation Experiments

A. Simulation Setup

For the experiments, we created a C-ITS environment using CARLA software [25] on an urban layout named *Town02*, covering an area of of 200×200 m^2. A view of this map is illustrated in Fig. 1. In this scenario, 20 connected automated vehicles (CAVs) are uniformly generated within the map's limits and navigate for $H = 1500$ time steps, with each step occurring every 0.2 seconds. In the scene, there are also 72 poles, which are detected by the CAVs if they fall within a direct line of sight and a sensing distance of 70 meters. This

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 $n = 1, \ldots, N_{\text{step}}$ **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 $t = 1$ to 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 (11)
      Compute $\bar{\boldsymbol{a}}_{i,t}$ and $\bar{\boldsymbol{h}}^{\text{b}}_{i,t}$ with (9) and (10)
      Observe $\boldsymbol{s}_{i,t+1}, \boldsymbol{o}_{i,t+1}$
      Get state estimate $\widehat{\boldsymbol{s}}_{i,t+1}$ with (7)
    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 $\ell = 0, \ldots, \lfloor H/L_\tau \rfloor$ **do**
    $\mathcal{B} = \mathcal{B} \cup \left\{\boldsymbol{\tau}_t, \widehat{A}_t, R_t\right\}_{t=\ell}^{\ell+L_\tau}$
    Adam update of $\boldsymbol{\psi}$ on $L(\boldsymbol{\psi})$ with data $\left\{\boldsymbol{\tau}_t\right\}_{t=\ell}^{\ell+L_\tau}$
  end for
  for each mini-batch **do**
    Sample $\left\{\boldsymbol{\tau}_\ell\right\}_{\ell=1}^{L_\tau} \sim \mathcal{B}$
    Adam update of $\boldsymbol{\theta}$ on $L(\boldsymbol{\theta})$ with data $\left\{\boldsymbol{\tau}_\ell\right\}_{\ell=1}^{L_\tau}$
    Adam update of $\boldsymbol{\phi}$ on $L(\boldsymbol{\phi})$ with data $\left\{\boldsymbol{\tau}_\ell\right\}_{\ell=1}^{L_\tau}$
  end for
  $\boldsymbol{\theta}_{\text{old}} = \boldsymbol{\theta}$, $\boldsymbol{\phi}_{\text{old}} = \boldsymbol{\phi}$
end for

sensing range is also consistent for A2A interactions. Moreover, we introduce Gaussian noise with a standard deviation of 2 meters to the GNSS, A2A, and A2T measurements.

We produced two different ground truth simulations for training and testing the ICP-MAPPO algorithm, while we generated unique noisy measurement realizations at each training and testing steps. Unless stated otherwise, we conduct 40 Monte Carlo (MC) evaluations during testing. The training process utilizes half the total time steps as the trajectory length, i.e., $L_\tau = H/2$, to facilitate the use of up to two mini-batches, following guidelines in [27], [29]. We selected the entropy and clipping coefficients of MAPPO as 0.01 and 0.2, respectively, while the reward coefficient was set to 0.05. We set the discount factor $\gamma = 0.99$ and the learning rate for the Adam optimization algorithm [30] to 10^{-5}, adhering to conventional settings.

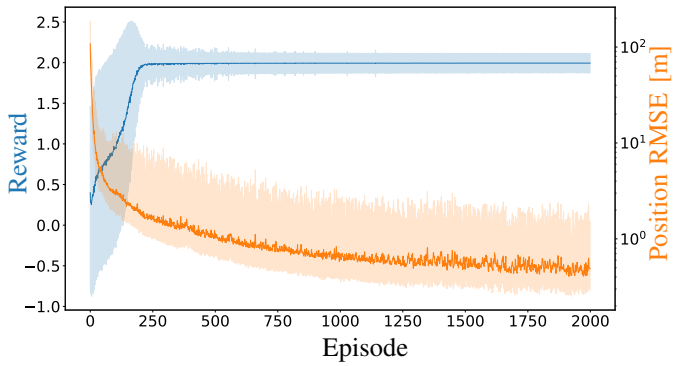
Regarding the NN architecture, the critic network comprises three layers: a fully-connected (FC) linear layer with 256 neurons, a gated recurrent unit (GRU) layer with a 256 neuron capacity, and a concluding FC linear layer. The actor network is an MLP with two hidden linear layers containing 128 and 64 neurons, respectively, employing rectified linear unit (ReLU) activation, and a sigmoid-activated output layer. The belief network incorporates two bidirectional LSTM layers, each with 256 hidden neurons and ReLU activations, a Maxout unit producing 128 output features, and two linear layers of 64 and 32 neurons.

In our study, we compared the proposed ICP-MAPPO algorithm with two key baseline algorithms. We implemented an EKF-GNSS, meaning a non-cooperative, single-agent GNSS-based EKF that relies solely on GNSS observations. The filter uses in the tracking model the same measurement uncertainties as in generation. Specifically, the Gaussian-distributed measurement noise terms of the measurements have the following covariance matrices $\boldsymbol{R}^{(\text{GNSS})}_{i,t} = \boldsymbol{R}^{(\text{A2A})}_{i,j,t} = \boldsymbol{R}^{(\text{A2T})}_{i,k,t} = 4\boldsymbol{I}_2$ m^2, For the vehicle motion model, a constant velocity model is assumed, Gaussian-distributed driving process $\mathbf{w}^{(\text{A})}_{i,t}$, calibrated with a standard deviation of 0.5 m/s^2. We also implemented a centralized ICP method [10], which uses the true standard deviations for both A2A and A2F measurements. This approach shares the same motion model as the EKF-GNSS. Notably, the ICP method assumes a fully-connected network of agents, meaning all agents have access to and share the same measurement data. This serves as a lower-bound on the performances of distributed ICP solutions. Moreover, the precise usage of measurement statistics in both the generation of scenarios and tracking processes enables an optimal evaluation of performance by minimizing errors that could arise from incorrect modeling.

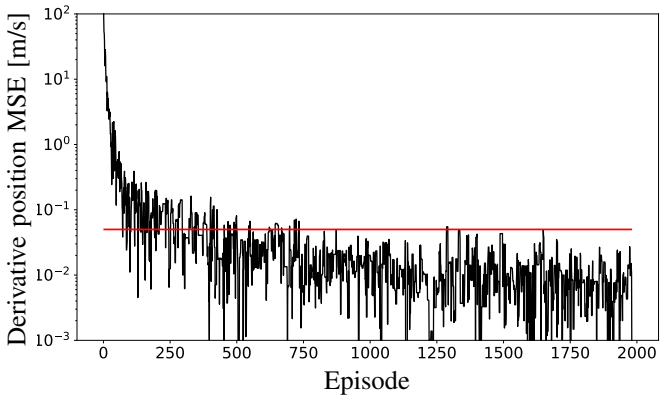
B. Simulation Results

1) Training convergence: In the first experiment, we verify the convergence of the proposed ICP-MAPPO algorithm along training episodes and the effect of the chosen reward coefficient β. To this aim, in Fig. 3a, we show the mean and 5-95 percentiles of the reward and root mean square error (RMSE) on the vehicle position, varying the number of training episodes. The mean and the error bounds are computed among the trajectory and agents. From the figure, we can notice that the mean reward passes from an initial phase of big ($> \beta$) negative and positive improvements, i.e., -1 and $+1$, to a convergence phase after about 250 episodes. To better explain this behaviour, we also report in Fig. 3b the derivative among timesteps of the MSE on the position (i.e., squared RMSE of the orange line in Fig. 3a). Note that the convergence happens when the derivative MSE falls below the $\beta = 0.05$ value, highlighted with a red line. Indeed, β physically translates to a reward function enhancement of β m in a scenario with non-standardized state space. Therefore β can regulate the trade-off between speed of policy convergence and accuracy of position estimates.

2) Impact of the number of detected targets: This assessment has the objective of quantifying the impact of the number of the detected targets on the positioning performances. Intuitively, the higher the number of targets simultaneously detected by two or more agents, the higher the number of



(a)



(b)

Fig. 3. (a) Mean reached reward (blue line) and RMSE on the position (orange line) varying the number of training episodes. The uncertainty areas are the 5-95 percentiles. (b) Derivative of the MSE on the position varying the number of training episodes (black line) and β threshold (red line).

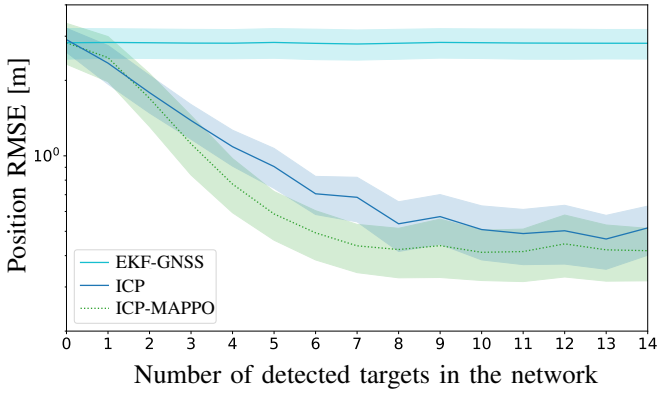


Fig. 4. RMSE on the position varying the maximum number of detected targets by each agent in the network.

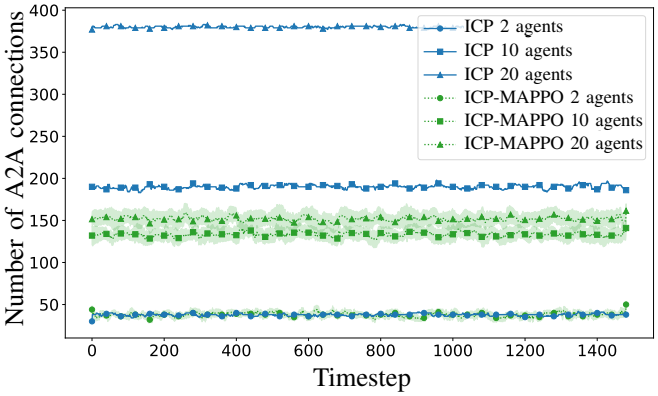


Fig. 5. Number of A2A connections in the network graph at each timestep, for the ICP and the proposed ICP-MAPPO algorithms, and different maximum connectivity in the graph.

anchors and the resulting positioning accuracy will be. Given that the maximum number of poles detected by two vehicles is 14, in Fig. 4, we show the RMSE on the position for the different methods by manually setting a maximum limit of detectable targets by each agent. From the figure, we first notice that the EKF-GNSS represents a lower bound on the performances when it comes to single agent stand-alone positioning with no measurements shared among agents. When cooperation is possible, we observe that ICP-MAPPO outperforms the Bayesian-filtering ICP solution, reducing the RMSE from 50 cm to about 41 cm. This improvement makes ICP-MAPPO suitable for positioning requirements on vehicles platooning in steady state and cooperative adaptive cruise control where the accuracy need to be lower than 50 cm [31], [32].

3) Impact of the number of cooperative agents: In this last experiment, the aim is to measure the communication efficiency obtained by ICP-MAPPO with respect to the ICP algorithm, where all agents are selected by the centralized implementation. Therefore, in Fig. 5, we report the cumulative number of A2A connections in the network varying the timestep of the trajectory and for different maximum allowed connectivity of each agent (i.e., 2, 10 and 20). We can notice that when the number of cooperative agents is low (e.g., 2) the ICP-MAPPO tends to select all available agents since neighbors' measurements can quickly reduce the GNSS uncertainty. On the contrary, when many agents are available, especially above 10, the cooperation becomes redundant, as only the nearest neighbors with a high number shared targets will have a major contribution in the positioning accuracy. We can see that, with 10 and 20 agents, ICP-MAPPO, in comparison to ICP, reduces the number of links of 30% and 60%, respectively.

V. Conclusion

In this study, we proposed a solution for CP in a distributed network of agents that utilize detected passive targets to enhance positioning accuracy following the ICP framework. We introduce a generalized ICP solution modelled as a Dec-POMDP, where the unknown agent state is estimated through histories comprising both measurements, i.e., GNSS, A2A and A2T observations, and A2A link activation, i.e., actions. The proposed ICP-MAPPO algorithm predicts the state with belief learning and dynamically optimizes the A2A

cooperation graph, or equivalently the communication links, with a refined version of the MAPPO algorithm.

Through realistic simulations in a C-ITS scenario using the CARLA environment, we demonstrate the superior performance of ICP-MAPPO over conventional ICP methods in terms of both positioning accuracy and communication efficiency. In particular, ICP-MAPPO better exploits the cooperative detection of targets and actively selects the best set of neighbors that give a relevant contribution to the positioning accuracy. Future works could extend the action decisions to not only A2A link selection, but also active A2A link communication improvement such as beamforming or packet scheduling.

REFERENCES

[1] T. G. Reid, S. E. Houts, R. Cammarata, G. Mills *et al.*, "Localization requirements for autonomous vehicles," *SAE Int. J. Connected Automated Veh.*, vol. 2, no. 3, pp. 12–02–03–0012, Oct. 2019.

[2] L. Italiano, B. Camajori Tedeschini, M. Brambilla, H. Huang *et al.*, "A tutorial on 5G positioning," *ArXiv*, Mar. 2024.

[3] 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.

[4] A. Conti, F. Morselli, Z. Liu, S. Bartoletti *et al.*, "Location awareness in beyond 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 22–27, Nov. 2021.

[5] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[6] Z. Liu, A. Conti, S. K. Mitter, and M. Z. Win, "Communication-efficient distributed learning over networks–part I: Sufficient conditions for accuracy," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1081–1101, Apr. 2023.

[7] Z. Liu, A. Conti, S. K. Mitter, and M. Z. Win, "Communication-efficient distributed learning over networks–part II: Necessary conditions for accuracy," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1102–1119, Apr. 2023.

[8] S. Fang, H. Li, and M. Yang, "Lidar SLAM based multivehicle cooperative localization using iterated split CIF," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21 137–21 147, May 2022.

[9] Y. Zhang, L. Chen, Z. XuanYuan, and W. Tian, "Three-dimensional cooperative mapping for connected and automated vehicles," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6649–6658, Sep. 2020.

[10] G. Soatti, M. Nicoli, N. Garcia, B. Denis *et al.*, "Implicit cooperative positioning in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3964–3980, Dec. 2018.

[11] 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.

[12] L. Barbieri, B. C. Tedeschini, M. Brambilla, and M. Nicoli, "Implicit vehicle positioning with cooperative lidar sensing," in *ICASSP 2023 - 2023 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[13] 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, Mar. 2024.

[14] M. Brambilla, D. Gaglione, G. Soldi, R. Mendrzik *et al.*, "Cooperative localization and multitarget tracking in agent networks with the sum-product algorithm," *IEEE Open J. Signal Process.*, vol. 3, pp. 169–195, Mar. 2022.

[15] 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.

[16] M. Liang and F. Meyer, "Neural enhanced belief propagation for cooperative localization," in *2021 IEEE Statistical Signal Process. Workshop (SSP)*, Jul. 2021, pp. 326–330.

[17] B. Camajori Tedeschini, M. Brambilla, L. Barbieri, G. Balducci *et al.*, "Cooperative lidar sensing for pedestrian detection: Data association based on message passing neural networks," *IEEE Trans. Signal Process.*, vol. 71, pp. 3028–3042, Aug. 2023.

[18] 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.

[19] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 895–943, Apr. 2021.

[20] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*, ser. SpringerBriefs in Intelligent Systems. Springer International Publishing, Jun. 2016.

[21] Z. Xia, J. Du, J. Wang, C. Jiang *et al.*, "Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 931–945, Nov. 2021.

[22] B. Peng, G. Seco-Granados, E. Steinmetz, M. Frohle *et al.*, "Decentralized scheduling for cooperative localization with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4295–4305, May 2019.

[23] M. Rangwala and R. Williams, "Learning multi-agent communication through structured attentive reasoning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 10 088–10 098.

[24] E. Pesce and G. Montana, "Learning multi-agent coordination through connectivity-driven communication," *Machine Learning*, vol. 112, no. 2, p. 483–514, Dec. 2022.

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez *et al.*, "CARLA: An open urban driving simulator," in *Conf. robot learning*. PMLR, Nov. 2017, pp. 1–16.

[26] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Auton. Agents Multiagent Syst.*, ser. Lecture Notes in Computer Science, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds. Springer International Publishing, Nov. 2017, pp. 66–83.

[27] C. Yu, A. Velu, E. Vinitsky, J. Gao *et al.*, "The surprising effectiveness of PPO in cooperative, multi-agent games," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, Mar. 2021, pp. 1–14.

[28] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan *et al.*, "Model-based value estimation for efficient model-free reinforcement learning," *ArXiv*, Feb. 2018.

[29] A. Ilyas, L. Engstrom, S. Santurkar, D. Tsipras *et al.*, "A closer look at deep policy gradients," in *Proc. 8rd Int. Conf. Learn. Representations*, Nov. 2020, pp. 1–27.

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

[31] 5GAA, "C-V2X use cases and service level requirements - volume II," 5GAA Automotive Association, Technical Report (TR), 2021.

[32] 5GAA, "C-V2X use cases and service level requirements - volume III," 5GAA Automotive Association, Technical Report (TR), 2023.