Multi-Agent Advantage Actor-Critic Learning For Message Content Selection in Cooperative Perception Networks

Imed Ghnaya, Mohamed Mosbah, Hasnaâ Aniss, Toufik Ahmed

Multi-Agent Advantage Actor-Critic Learning For Message Content Selection in Cooperative Perception Networks

Imed Ghnaya
Univ. Bordeaux, Bordeaux INP CNRS, LaBRI, UMR5800
F-33400 - Talence, France
imed.ghnaya@u-bordeaux.fr

Mohamed Mosbah
Univ. Bordeaux, Bordeaux INP CNRS, LaBRI, UMR5800
F-33400 - Talence, France
mohamed.mosbah @u-bordeaux.fr

Hasnaâ Aniss
Gustave Eiffel University COSYS-ERENA Lab
F-33067 - Bordeaux, France
hasnaa.aniss@univ-eiffel.fr

Toufik Ahmed
Univ. Bordeaux, Bordeaux INP CNRS, LaBRI, UMR5800
F-33400 - Talence, France
tad@labri.fr

***Abstract*—Recent advancements in autonomous vehicle perception have exposed limitations of onboard sensors such as radar, lidar, and cameras, which road obstacles and adverse weather conditions can impede. Connected and Autonomous Vehicles (CAVs) are leveraging wireless communications to share perception information through a process called Cooperative Perception (CP), aiming to provide a more comprehensive understanding of their environment. However, this can result in excessive redundant and useless information in the network, as the same road objects may be detected and exchanged simultaneously by multiple CAVs. This not only consumes more network resources but also may overload the communication channel, reducing the delivery of perception information to CAVs and ultimately decreasing the overall CP awareness in the network. This paper introduces MCORM, a multi-agent learning method based on the advantage actor-critic algorithm to maximize object usefulness and reduce redundancy in the network. Our evaluations demonstrate that through this method, CAVs learn optimal CP message content selection policies that maximize usefulness. Furthermore, our proposal proves to be more effective in mitigating object redundancy and improving network reliability in comparison to existing approaches.**

Keywords—connected and autonomous vehicles, cooperative perception, redundancy mitigation, multi-agent system, reinforcement learning, advantage actor-critic.

I. INTRODUCTION

One of the paramount challenging encountered in autonomous driving pertains to environmental perception. Vehicles perceive their surroundings (e.g., other vehicles, obstacles, and pedestrians) using onboard sensors, such as radars, lidars, and cameras, which enable Advanced Driving Assistance Systems (ADASs) to furnish road users with better comfort and safety services. Nevertheless, the capability of each sensor is constrained by its limited detection range and Field-of-View (FoV), which can be hindered by the presence of obstructions on the road, inclement weather conditions, and other environmental factors. These limitations can significantly impair vehicles' perception capabilities, thereby compromising their safety and performance [1].

Recent improvements in Vehicle-to-Everything (V2X) communications [2] provide viable alternatives to this limitation by enabling connected road users to share information using wireless communication technologies. In this regard, the European Telecommunications Standards Institute (ETSI) has standardized the ITS-G5 [3] as an IEEE 802.11p-based communication protocol, especially for Connected Autonomous Vehicles (CAVs) to exchange vital information, such as speed, position, and heading, via Cooperative Awareness Messages (CAMs) [4].

The Cooperative Perception (CP) service [5] is utilized to enhance road safety and increase the environmental awareness of CAVs. Specifically speaking, the CP may essentially be divided into V2V and Vehicle-to-Infrastructure (V2I). In the former case, CAVs directly exchange their sensory information via periodic broadcasting, allowing for more timely utilization of crucial data in making critical driving decisions. In the V2I-based CP, CAVs periodically send their sensory information to a roadside infrastructure using V2I communications for further processing. In both cases, the communication is achieved through message exchange, where sensory information is included in Cooperative Perception Messages (CPMs) as high-level descriptions of tracked objects, such as speed, position, height, and width. As this paper primarily focuses on V2V-based CP, the same objects in the driving environment can appear simultaneously in the Field of View (FoV) of multiple CAVs. As a result, their exchange through CP may result in a significant amount of redundant information being transmitted in the V2V network, potentially degrading network reliability by increasing channel load and reducing CPM delivery, thereby lowering the CP awareness level in the V2V network.

Given the ever-changing nature of the driving environment, the sharing of CPMs among CAVs must be automatic and online to provide timely and relevant perception data. All possible combinations of perceived objects must be assessed for the transmitter CAV to determine the best CPM. However, this process can be challenging due to the high computational complexity and impracticality of evaluating all possible combinations of perceived objects to determine the best CPM in dense driving scenarios. Reinforcement Learning (RL) [6] can be an effective solution to this problem as it allows a CAV to learn which objects to include in a CPM based on the state of its surroundings, with the aim of maximizing a reward value. However, traditional RL training using value-based RL

algorithms, such as Q-learning and DQN [6], in large state and action spaces seems impossible due to the complexity of the environment [7]. Policy-based RL algorithms [6], such as policy gradient, gradually fit and evaluate a policy without exploring entire spaces in order to overcome this limitation to some extent. However, they tend to generate a large variance in estimating the gradient since they are usually updated per round, which reduces the training efficiency. An Actor-Critic (AC)-based RL algorithm [8] was proposed to boost RL-based systems in complex problems by taking the strengths of both the value-based and policy-based methods. In fact, AC employs linear value functions typically represented by Deep Neural Networks (DNNs) to approximate the action-value function. However, authors in [9] demonstrated that AC generates a high variance and gives inaccurate outputs. In [10], an Advantage AC (A2C) algorithm is introduced to reduce training variation. Using its policy, the actor chooses an action that is then evaluated by the critic, who returns an advantage value. This value, indicating the value of the chosen action, enables the actor to adjust its policy accordingly. As our paper addresses the CP in a multi-CAV environment, we develop a Multi-CAV A2C framework to address the dynamicity of each CAV independently. In this area, the authors of [11] have focused on the Centralized Training for Decentralized Execution (CTDE) paradigm in order to build a multi-agent actor-critic environment that serves as a Centralized Critic for Decentralized Actors (CCDA).

This paper proposes a multi-agent advantage actor-critic learning method, namely MCORM, for a multi-CAV driving environment. Its primary objective is to enable each CAV to learn a CPM content selection policy that maximizes the receiving CAVs' usefulness as much as possible to reduce the number of redundant objects in the V2V network. Results show the ability of MCORM to adapt to the driving environment and to learn CPM content selection while maximizing usefulness effectively. Furthermore, the results highlight that the proposal effectively mitigates object redundancy and improves network reliability without static thresholds. This ensures an increased awareness of CAVs compared to state-of-the-art approaches.

The main contributions of this paper are summarized as follows:

- We suggest a mathematical approach to optimize the usefulness of information received from onboard sensors and V2V communications in a multi-CAV environment. This approach considers various perception factors, such as distance, object size, viewing angle, and obstructions caused by other road users, to adapt as effectively as possible to the driving environment.
- CAVs may not have complete information about their surroundings due to limitations from onboard sensors and network restrictions. To address this issue, we use a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) framework [12] to characterize the uncertain information in MCORM's environment. Additionally, a position-based representation may not be sufficient for the training process as CAVs constantly move and update their decision-making process. To address this, we propose a scalable design for the state and action spaces of the Dec-POMDP, allowing CAVs to leverage previous experiences in various regions and at any time.
- We introduce the A2C learning algorithm in a multi-CAV environment based on CCDA to solve the CPM content selection problem on each CAV. Specifically, each CAV acts as an agent, using a DNN for its policy function to select the perceived objects that maximize its neighbors' usefulness based on the state of its environment. On the other hand, we design a DNN model to characterize the value function representing the critic in a central computing and storage device. Agents representing CAVs employ V2D communications to exchange training-related information with the central device and V2V communications to exchange CAMs and CPMs with other agents (CAVs).
- We implement and evaluate MCORM based on the PyTorch library and through advanced discrete-event network and road traffic simulators Artery [13] and SUMO [14], respectively. We then present results and evaluations to show its performance compared to the state-of-the-art approaches.

The rest of this paper is organized as follows. In Section II, we provide an overview of the most recent related works. The problem formulation of CP's information usefulness is presented in Section III. In Section IV, we introduce the MCORM design and learning algorithm. Simulations and evaluations are presented in Section V. Finally, we conclude this paper in Section VI.

II. Related works

The ETSI has proposed a set of CPM generation rules to balance the channel load and the amount of perception data to be exchanged in the V2V network [5]. These rules specify whether a CAV generates and broadcasts a CPM and what information it should include based on periodic and dynamic policies. The periodic policy generates CPMs periodically at every generation interval. In every CPM, the transmitting CAV includes information about all detected objects. The CPM should be transmitted even if no objects are detected. The periodic policy is being used as a benchmark in the standardization process to compare its performance and efficiency with more advanced policies such as the dynamic one. With the dynamic policy, the transmitting CAV checks if the environment has changed and if it is necessary to generate and transmit a new CPM at every generation interval. A CAV generates and broadcasts a CPM if one of the following conditions is satisfied. (1) It detects a new object. (2) Its position or speed has changed by 4 m (meters) or 0.5 m/s (meters per second), respectively, since the latest information included in its CPM. (3) The last time the detected object was included in a CPM was 1 (or more) seconds ago. If none of the above conditions are satisfied, the CAV still generates a CPM every 1 s.

According to a recent study in [15], the CPM generation rules result in significant information redundancy in the V2V network. This is because CAVs do not analyze perceived information from other CAVs in their environments. A recent dynamics-based redundancy mitigation technique is proposed in

[16], where each CAV analyzes the most recent CPMs received from other CAVs and excludes perceived objects that exceed predefined position or speed thresholds. In addition, authors in [17] have proposed redundancy control schemas based on channel status, number, and type of V2X stations that have also provided the same perceived information. The main objective is to adapt the number of V2X stations transmitting data about the same object to the channel load while maintaining CP awareness close to the default CPM generation rules. However, these techniques consider predefined and static thresholds, which may not have the appropriate settings in heterogeneous driving environments with varying situations and vehicular densities. Moreover, the authors in [18] have demonstrated that the existing message generation rules may produce a high level of redundant information in highway scenarios and then theoretically proposed a probabilistic data selection scheme [19]. This schema allows each CAV to adjust the adaptive transmission probability for each detected object based on its position and road traffic information. In contrast, the communication part of this technique is evaluated through MATLAB instead of network simulators such as Artery, which simulate communications using the ETSI ITS-G5 protocol stack.

In summary, most existing techniques focus on predefined thresholds to include perceived objects in a CPM based on their position or speed criteria. There still lacks a consideration of information usefulness over the coverage of the transmitter CAV. To our knowledge, [20] is the first and only work that has proposed omitting redundant objects based on object usefulness. The authors of this article have introduced a deep RL-based schema where object usefulness is modeled as a reward based on only the distance from the perceived object to the CAV receiving it. Nonetheless, this modeling does not consider various perception contexts, such as object size and road occlusions, which may influence the usefulness model and affect the CP awareness level in the V2V network.

III. Problem Formulation

This section formulates the usefulness of information perceived from onboard sensors and V2V communications in a multi-UAV environment as a maximization problem. This formulation takes several perceptual factors, such as position, distance, object size, viewing angle, and occlusions, to adapt as much as possible to the environment. Table I provides a summary of the notations used in this work.

We freeze the image of the driving environment depicted in Fig. 1 at time t. At this time, we define the driving environment, $V(t) = \{v_1(t), v_2(t), \dots, v_n(t)\}$, as a set consisting of n CAVs indexed from 1 to n. Each CAV is represented as a rectangle, $v_i(t) = (c_i(t), l_i, w_i)$, where $c_i(t)$ is the geometric center of the rectangle represented by its X-position $x_i(t)$ and Y-position $y_i(t)$ at time t on a global 2-dimensional plane, its length l_i and width w_i. As an assumption, all CAVs in the driving environment have identical capabilities, as follows:

- Each CAV is fitted with Global Positioning System (GPS) and Global Navigation Satellite System (GNSS) devices to provide timely information about its position.
- Each CAV is equipped with 360° sensors, such as radars and lidars, to perceive road objects. We define the 360° sensing coverage as a circle whose radius m representing the maximum sensing range across all sensors.
- Each CAV is fitted with V2V and V2D wireless communication devices to share information with other CAVs and the central device.

According to the above assumption, Fig. 2 illustrates the principle internal services related to this work for a given CAV. Each CAV performs a Cooperative Awareness Service every t =100 milliseconds (ms) [4]. This service gathers information from GPS/GNSS and other sources to generate and broadcast status information via CAMs [4] to the other CAVs through the V2V Tx Interface. The status information of a CAV $h_i(t)$ can

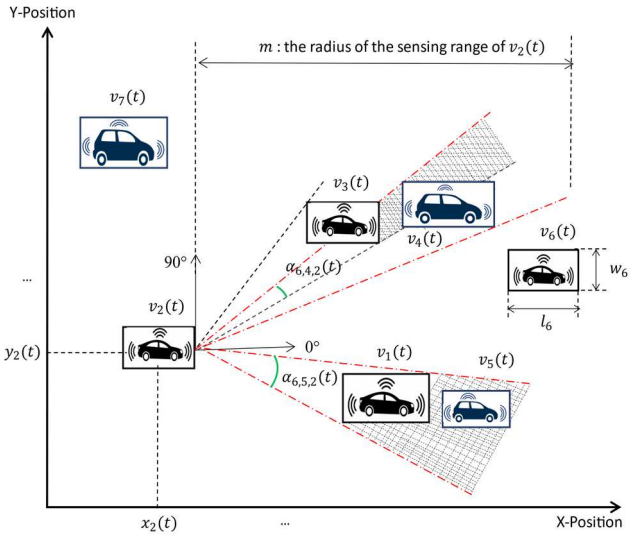

Fig. 1. A geometric representation of the driving environment at time t.

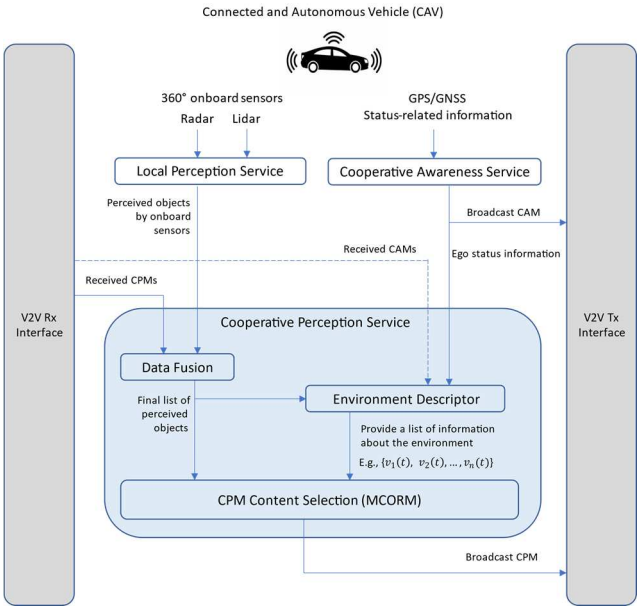

Fig. 2. In-CAV Services.

be described by one or a set of features, including its position, speed, length, and width. To simplify, we only consider the position on a global 2-dimensional plane, length, and width of each CAV. Thus, we express, $h_i(t) = \{x_i(t), y_i(t), l_i, w_i\}$, as a set of features consisting of $x_i(t)$, $y_i(t)$, l_i, and w_i. Following the exchange process, each CAV receives other status information, $H_i(t) = \{h_j(t), j \neq i\}$, from other CAVs through the V2V Rx Interface.

We consider another time step t' when CAVs perceive and exchange objects through CPMs via V2V communications as follows. First, each CAV starts by performing a Local Perception Service to perceive the road environment using its onboard sensors. Then, it conducts a Cooperative Perception Service that consists of three principal modules. The Data Fusion Module fuses locally perceived objects with received objects through the V2V Rx Interface from other CAVs. This module outputs the final list of objects to be included in a CPM and broadcast to other CAVs. We define this final list, $T_i(t') = \{o_{i,1}(t'), o_{i,2}(t'), \dots, o_{i,k}(t')\}$, as a list of perceived features, such as position, length, and width for each object *j* perceived by the *i*th CAV as follows, $o_{i,j}(t') = \{x_j(t'), y_j(t'), l_j, w_j\}$. The exchange of perceived objects between CAVs without any prior intelligence could result in a significant amount of useless and redundant information in the V2V network. Regarding this limitation, we propose a CPM Content Selection Module, which enables each CAV to select and broadcast only useful objects that maximize the benefit of its surrounding CAVs. The selection process doesn't employ static thresholds and adapts to the situation of the driving environment. This is done by employing the information provided by an Environment Descriptor Module, which exploits the received CAMs and CPMs to create a geometric representation of the surrounding elements in the driving environment at time step t' (e.g., provides as an output a list of the position, length, and width for each surrounding element).

To that end, the maximization problem at the *i*th CAV is formulated as follows:

$$\underset{P_i(t')}{\text{maximize}} \quad 1 - \left(\frac{1}{n'n''} \sum_{\substack{k=1 \\ k \neq i}}^{n'} \sum_{\substack{j=1 \\ j \neq k}}^{n''} f_{i,j,k}(t') * g_{i,j,k}(t') \right), \qquad (1)$$

Subject to:

$$t \leq t' < 2t, \qquad (2)$$

$$f_{i,j,k}(t') = \begin{cases} 0, & d_{i,j,k}(t') > m \\ 1 - \frac{d_{i,j,k}(t')}{m}, & otherwise \end{cases} \qquad (3)$$

$$g_{i,j,k}(t') = \begin{cases} 0, & \phi_{i,j,k}(t) = \sum \alpha_{i,j,k}(t') > \rho_{i,j,k}(t') \\ 1 - \frac{\phi_{i,j,k}(t')}{\rho_{i,j,k}(t')}, & otherwise \end{cases} \qquad (4)$$

$P_i(t') = \{o_{i,1}(t'), o_{i,2}(t'), \dots, o_{i,n''}(t') \; ; \; o_{i,j}(t') \in T_i(t')\}$ is the list of n'' useful objects to be included in a CPM. n' is the number of CAVs in the communication coverage of the *i*th CAV. Equation (2) ensures that CAVs broadcast and receive CAMs before sharing CPMs and that the environment is unchanged between t and t'. Equation (3) computes a distance-related factor $f_{i,j,k}(t')$ in [0,1] that refers to the value of which the jth object $o_{i,j}(t')$ perceived by the ith CAV appears in the sensing coverage of the kth CAV as follows: the closer the Euclidian distance $d_{i,j,k}(t')$ between $o_{i,j}(t')$ and the kth CAV is to m, the closer the factor is to zero, meaning that $o_{i,j}(t')$ is being unperceivable by the kth CAV, thereby being useful for it. Equation (4) denotes an occlusion-related $g_{i,j,k}(t')$ in [0,1] that refers to the value of which $o_{i,j}(t')$ is directly in the LoS of the kth CAV. This factor is defined proportionally to the sum $\phi_{i,j,k}(t')$ of all angles (e.g., $\alpha_{i,j,k}(t)$) that overlap and occlude the viewing angle $\rho_{i,j,k}(t')$ from the kth CAV to $o_{i,j}(t')$. This means that the closer this sum is to the viewing angle, the more the LoS to this object is occluded. As an example, for $i = 6,\ j = 4,$ and $k = 2$ in Fig. 1, $\alpha_{6,4,2}(t')$ represents the only occlusion angle that occludes the viewing angle $\rho_{6,4,2}(t')$ from 2nd CAV $v_2(t)$ to the 4th CAV $v_4(t')$, which makes the 2nd CAV cannot successfully detect this latter. We employ the $atang2$ function to determine the angle between two positions in a global 2D coordinate system. Since $atang2$ computes the angle between a given position and the X-axis, a simple subtraction can be performed to get the recommended angle.

IV. Advantage Actor-Critic Learning in a Multi-CAV Setting

This section provides the system model and learning algorithm to mitigate redundant perception information in the V2V network.

A. System design

Agents in MARL interact with a stateful environment to solve a sequential decision-making problem. Each agent takes action based on the state of the environment and then receives feedback at each timestep to maximize its reward. Typically, a fully observable environment is modeled as an MDP, where each agent collects complete state information. However, in our multi-CAV-based driving environment, CAVs representing agents build the state of their environments using information perceived by onboard sensors and received by CAMs and CPMs. Due to limited onboard sensors and network constraints, CAVs may not perceive complete information about their surroundings, making the state of their environments uncertain and only partially observed. For that reason, we build a Decentralized Partially Observable MDP (Dec-POMDP) framework [12] to characterize MCORM's environment states under uncertain information. A DEC-POMDP at the *i*th CAV is mainly defined by its essential components as follows:

Environment state: Given that the CAVs move and store environment state and action pair each time to improve their decision-making process, a position-based presentation cannot be used to conduct the training process. Therefore, we introduce a scalable position-independent representation of the environment state that allows the *i*th CAV to leverage past state-action experiences:

$$s_i(t') = \{(d_{i,j}(t'), \beta_{i,j}(t'), l_j, w_j); \forall\, j \neq i\}, \qquad (5)$$

TABLE I. NOTATION SUMMARY

Notation	Description	Notation	Description
t, t'	Status-related and perception-related time instances	m	Sensing range
i, j, k	Indexes	$\rho_{i,j,k}(t')$	The viewing angle from the kth CAV to $o_{i,j}(t')$.
$x_i(t), y_i(t), l_i, w_i$	The position,length, and width of the ith CAV	$\alpha_{i,j,k}(t')$	An occlusion angle that overlap and occlude $\rho_{i,j,k}(t')$
$v_i(t)$	The rectangle representing the ith CAV	$\phi_{i,j,k}(t')$	The sum of all occlusion angles overlap and occlude $\rho_{i,j,k}(t')$
$V(t)$	The set of n CAVs representing the driving environment	$s_i(t'), z_i(t')$	The state and observation of the ith CAV
$h_i(t)$	The status information of the ith CAV	$d_{i,j}(t'), \beta_{i,j}(t')$	The distance and viewing angle from the ith CAV to the jth CAV.
$H_i(t)$	The set of status information received by the ith CAV from other CAVs	p, s	The number of pistes and sectors
$o_{i,j}(t')$	The jth object perceived by the ith CAV	$FoV_i(t')$	The vector of the $p * s$ FoV cells of the ith CAV
$T_i(t')$	The final list of perceived objects	$C^j_{p',s'}(t')$	The cell of $p'th$ piste and $s'th$ sector indexed by $j \in [0, p * s - 1]$ in $FoV_i(t')$
$P_i(t'), n''$	The list of n'' useful objects to include in a CPM	$a_i(t'), r_i(t')$	The action and reward of the ith CAV
$d_{i,j,k}(t')$	The Eclidean distance between the ith CAV and $o_{i,j}(t')$	Q, θ'	The Critic network parametraized by a vector of wheigts θ'
$f_{i,j,k}(t'), g_{i,j,k}(t')$	Distance-related and occlusion-related factors	π, θ	The Actor network parametraized by a vector of wheigts θ'

where $d_{i,j}(t')$ and $\beta_{i,j}(t')$ are the distance and viewing angle from the ith CAV to the jth CAV. l_j and w_j are the length and width of the jth CAV. As the environment is partially observable, we define an observation $z_i(t') \subseteq s_i(t')$ for the ith CAV as a partial set from $s_i(t')$ determined by its Environment Descriptor Module.

Action. Given the mobility of CAVs in the driving environment, the perception changes over time. Therefore, the action should be independent of changing characteristics. To that end, we propose a cell-based scheme that divides the circular FoV of each CAV into p pistes and s sectors. Hence, the FoV of the ith CAV at time step t' can be represented by a vector of cells of size $s * p$ as follows:

$$FoV_i(t') = \left[C^0_{1,1}(t'), \dots, C^1_{1,s-1}(t'), C^2_{1,0}(t'), \dots, C^{p*s-1}_{,p-1,s-1}(t')\right], \quad (6)$$

where $C^j_{p',s'}(t')$ is representing the cell of $p'th$ piste and $s'th$ sector indexed by $j \in [0, p * s - 1]$. To that end, we define the action space by the power set of $FoV_i(t')$ in order to cover all the unique possible combinations of cells with a complexity of $\Theta(2^{p*s} - 1)$. The action $a_i(t')$ can be a natural number in the range $[0, 2^{p*s} - 1]$. This number is then converted to a binary string $\left(b_0 b_1 \dots b_{p*s-1}\right)_2$ of length $p * s$. Following that, the objects that appear in $C^j_{p',s'}(t')$ will be included in CPM only if $b_j = 1$. The CPM include all perceived objects if all bits are set to 1s. On the other hand, the CPM is empty if all bits are set to 0s.

Reward. The reward $r_i(t')$ of the ith CAV at timestep t' is the usefulness of the objects generated from the selected action over its communication coverage. The reward function is denoted in (1).

B. Learning Algorithm for CAVs

Mainly, RL algorithms can be classified into two categories: policy-based and value-based. The policy-based algorithms, such as policy gradient [6], conduct an agent to learn a policy function representing a probability distribution over the action space. This function maps each environment state to an action to perform. Value-based algorithms, on the other hand, such as Q-learning and DQN [6], learn an agent to select actions based on the predicted value of the state or action. However, due to the dynamicity of the environment, which generates large state and action spaces, value-based algorithms are limited in conducting the learning process of CAVs in MCORM. In contrast, policy-based algorithms appear to perform better in this context, as they gradually fit a policy without exploring the state and action spaces but still generate a high variance in estimating the gradient value. The work in [10] introduced an A2C algorithm that combines advantages from both policy-based and value-based algorithms to reduce variance during the training process.

In this paper, we develop a distributed A2C-based learning algorithm for a multi-CAV environment to learn each CAV an optimal CPM content selection policy that maximizes the benefit of the receiving CAVs in the V2V network. In this context, the authors of [11] have introduced a centralized critic for decentralized actors to build a multi-agent actor-critic framework for mixed cooperative and competitive environments. Motivating by that, we adopt the same framework to develop the following Algorithm 1. The learning process requires the number of learning episodes $N_{episodes}$ and the number of steps per episode N_{steps} as inputs. Before starting the learning process, the central computing and storage device initializes a critic network Q with random parameters θ' and a buffer of experiences $\mathcal{D}$. Then, each CAV initializes a policy network π with random parameters θ (Lines 1-9).

Algorithm 1: The multi-CAV A2C learning algorithm for object redundancy mitigation in the V2V network

1: **Inputs:**
2: Number of learning episodes $N_{episodes}$
3: Number of steps per one learning episodes N_{steps}
4: **Output:**
5: Learned policy for each CAV
6: **Begin**
7: Initialize a critic network Q with random weights θ'
8: Initialize a buffer of experiences $\mathcal{D}$
9: Each CAV on creation initializes a policy network π with random weights θ
10: $episodes \leftarrow 0$
11: **while** $episodes < N_{episodes}$ **do**
12: $t \leftarrow 0$
13: **while** $t < N_{steps}$ **do**
14: **Each** available CAV i :
15: Builds an observation $z_i(t)$
16: Selects an action $a_i(t)$, generates and broadcasts a CPM
17: Gets a reward $r_i(t)$
18: Stores $e_i(t) = \big(z_i(t), a_i(t), r_i(t), z_i(t+1)\big)$ into $\mathcal{D}$
20: $t \leftarrow t + 1$ // Increase the steps by one CPM gen. interval
21: **end while**
22: Sample a minibatch from $\mathcal{D}$
24: Update θ' according to (7)
25: Update all CAVs' policies according to (9)
26: $episodes \leftarrow episodes + 1$
27: **end while**
28: **End**

The learning process is conducted $N_{episodes}$ episodes. During each episode, each CAV performs N_{steps} times the following four processes. First, it builds an observation $z(t')$ based on the information provided by its Environment Descriptor Module. Second, it chooses an action $a(t')$, generates objects, and broadcasts a CPM to other CAVs using the V2V communication. Next, it receives a reward $r(t')$ and sends an experience $e(t') = (z(t'), a(t'), r(t'), z(t'+1))$ to the central computing and storage device using V2D communication. This latter stores each received experience in a buffer of experiences shared for all CAVs (Lines 7-21). Following each N_{steps}, the critic updates its model first, and then each CAV updates its policy model before the system starts the next learning episode (Lines 22-28).

The critic model on the central computing and storage device is updated by minimizing the Temporal Difference (TD) calculated between the estimated and the actual values on a sampled minibatch of experiences $(s, a, r, s') \sim U(\mathcal{D})$ of size n_e drawn uniformly from the buffer of experiences $\mathcal{D}$, where s is the state, a is the chosen action, r is the reward, and s' is the next state. The update equation of the critic is denoted by:

$$\theta' = \theta' + \alpha * Loss(\theta'), \tag{7}$$

where α is the learning rate used to adjust the model parameters and

$$Loss(\theta') = (1/n_d) \sum_{t=0}^{n_e-1} \Big(r(t) + \gamma Q_{\theta'}\big(s(t+1)\big) - Q_{\theta'}\big(s(t), a(t)\big)\Big)^2, \tag{8}$$

is the critic loss. $\gamma \in [0,1]$ is a discount factor given to early values to reduce their impact.

On the hand, the update of the actor policy at the ith CAV is performed as follows:

$$\theta_i = \theta_i + \alpha * \nabla_{\theta_i} J(\theta_i), \tag{9}$$

where,

$$\nabla_{\theta_i} J(\theta_i) = \sum_{t=0}^{|\tau_i|-1} \nabla_{\theta_i} log\, \pi_{\theta_i}\big(a_i(t)|s_i(t)\big) A\big(s_i(t), a_i(t)\big), \tag{10}$$

is the gradient calculated based on the policy gradient method [6]. $\tau_i = [s_i(0), a_i(0), s_i(1), (a_i(1)s_i(2), a_i(2), \dots]$ is the local history of state-action of the ith CAV. The advantage value

$$A\big(s_i(t), a_i(t)\big) = r_i(t) + \gamma Q_\theta\big(s(t+1)\big) - Q_\theta\big(s(t), a_i(t)\big) \tag{11}$$

denotes how good the chosen action $a_i(t)$ is in the state $s_i(t)$.

V. Performance Evaluation

In this section, we evaluate the performance of MCORM based on simulations using Artery [13] and SUMO (Simulation of Urban Mobility) [14] simulators. We consider a 10 km² of a real-world obtained from OpenStreetMap and comprises different road traffic scenarios, including the city center and highways with different situations, such as ramps and T-junctions. On this map, we randomly generate vehicles with different types and sizes. We set up a GPS and 360° radar and lidar sensors for each CAV with a maximum sensing range of 100 m. Each CAV implements cooperative awareness and cooperative perception services to exchange CAMs and CPMs with other CAVs every 0.1s and 0.15s, respectively, using a 6Mbps data rate. We consider the ETSI ITS-G5 as a V2V communication profile for CAVs with a communication coverage of 500 m, where they can broadcast and receive messages through the control channel (CCH). we also install a central computing and storage device accessible for all CAVs in the driving scenario. We implemented MCORM based on the PyTorch library. We define a Multilayer Perceptron (MLP) network on each CAV to represent its actor policy function. We also build a MLP network to represent the critic value function on the central device. The simulation spans 60000 time slots, where each time slot represents a CPM generation interval. The training phase consists of $N_{episodes} = 4000$ learning episodes, where each episode be made up every $N_{steps} = 10$ CPM generation intervals. Furthermore, to conduct the training process, we consider a learning rate $\alpha = 10^{-3}$, a minibatch size of 64, a discount factor $\delta = 0.99$, and a buffer size $|\mathcal{D}| = 10^6$. Finally, for complexity reasons, we divide the FoV of each CAV into 3 pistes and 3 sectors, resulting in 9 distinct cells.

The aim of Algorithm 1 is to learn CAVs to select and broadcast perceived objects that maximize the usefulness of their surroundings. We start by studying the training convergence of our method in terms of reward variation. Fig. 3 illustrates the variation of the average reward as a function of learning updates. As seen, CAVs continually update their policies until they exceed the maximum number of training episodes ($N_{episodes} = 4000$). At the beginning of the learning process, the average reward showed large variations because the environment was changing and the CAVs lacked sufficient training experience to minimize their prediction loss efficiently.

However, we observe that this metric increases with the number of learning episodes and stabilizes after about 3500 update steps, indicating that CAVs maximize the usefulness of their CPMs in the V2V network after about 35000 time slots.

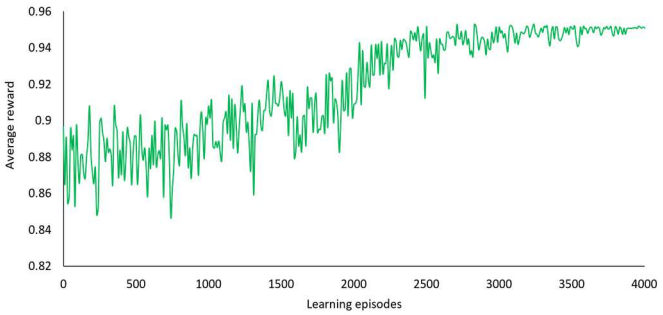

Fig. 3. The variation of the average reward as a function of learning episodes.

Following that, we compare the performances achieved by MCORM to the ETSI CPM generation rules [5] summarized at the beginning of Section II, dynamics-based [16] technique, and CBR-selective [17] scheme. A CAV utilizing the dynamic-based technique verifies the most recent CPM received from all neighboring CAVs and excludes perceived objects whose positions or speeds exceed static thresholds. However, using the CBR-selective scheme, the CAV selects objects based on the network state and the number of other CAVs that provide the same information about these objects. The CAV includes only perceived objects that do not exceed static CBR and redundancy thresholds. To compare performances, all approaches are performed in each time slot. We consider the same values defined in [5] for object redundancy for all approaches. An object is redundant if the difference in its absolute speed or position is less than 0.5 m/s and 4 m, respectively.

To show the performances offered by MCORM compared to the above-described works, we defined the following network-related KPIs:

- Object redundancy (OR): indicates the number of times a CAV receives identical information about the same perceived object over the selected time interval.
- Cooperative Perception Awareness (CPA): represents the number of unique objects known to a CAV, given the total number of objects in its coverage. We consider an object to be known by a CAV if it is successfully detected or received via V2V communication.
- CBR: identifies the current utilization percentage of the V2V communication channel. It is determined by assessing the channel in a time interval. Given the duration of one OFDM symbol 8 µs (48 bits per symbol at a data rate of 6 Mbps in the ITS-G5-CCH), the channel is assessed for $N = 12500$ symbols. Whenever the received signal strength exceeds -85 dBm, the channel is assessed as busy for this symbol. Thus, we define CBR as N_{busy}/N.
- CPM delivery (CDR): identifies the probability of correctly receiving a CPM at a given distance d to the CAV sender. Mathematically, the CDR at *i*th CAV at d is defined by $\sum_{j=0}^{N_d} \omega_{i,j}(d) \,/\, \sum_{j=0}^{N_d} \omega'_{i,j}(d)$, where $\omega'_{i,j}(d)$ is the number of CAVs whose Euclidean distances to the *i*th CAV are less than d when this latter transmits a CPM j. $\omega_{i,j}(d)$ is the number of CAVs that successfully receive the CPM j. N_d is the number of CPMs sent by the *i*th CAV. We define the PDR at d as an average CDR computed for all CAVs in the driving scenarios.

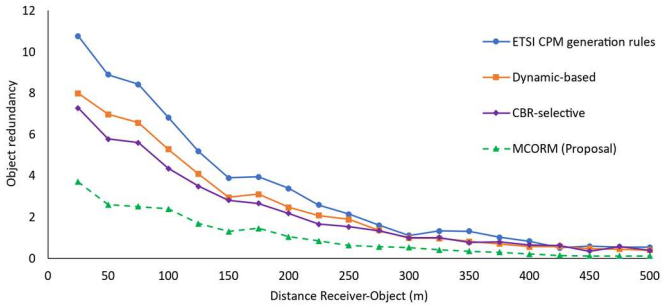

Fig. 4. OR as a function of the distance between the perceived object and the CAV receiving it

We start recording statistics for all network-related KPIs after the training process of the proposal reaches the total number of learning episodes (i.e., $N_{episodes} = 4000$ episodes). Fig. 4 depicts OR generated by the proposal and the other approaches. This KPI is plotted as a function of the distance between the perceived object and the CAV receiving it. We notice that the OR results in high levels at short distances because perceived objects are successfully detected and exchanged by multiple CAVs simultaneously. However, OR reduces with increasing distance because objects become farther away from the receivers, making their perception more challenging. Fig. 4 shows that ETSI CPM generation rules generate higher OR levels. This can be explained because CAVs generate and broadcast CPMs without prior intelligence. However, the dynamic-based technique and the CBR-selective scheme have a marginally reduced OR compared to the latter. The proposal reduces OR considerably at short and medium distances of less than 300 m. For instance, MCORM, on average, decreases OR by around 7 redundant objects compared to the ETSI CPM generation rules at short ranges of less than 50 m. However, this gain is lowered to around 4 redundant objects compared to the remaining approaches at the same distances.

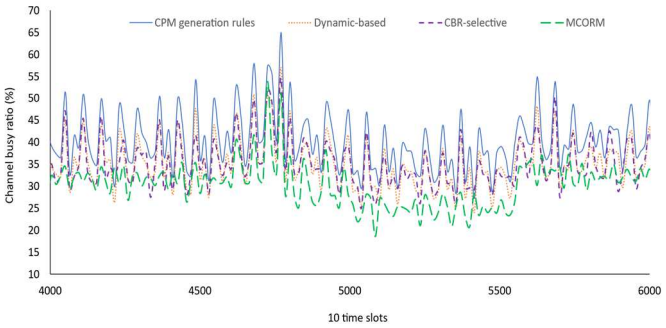

Fig. 5. The CBR as a function of time slots

The main purpose of mitigating OR is to reduce the resources used to disseminate useless perceived objects. Fig. 5 depicts the average CBR variation per 10-time slots for the proposal and other approaches. We observe that the ETSI CPM generation rules result in the highest CBR variations across time slots. However, the dynamic-based technique and the CBR-selective scheme perform marginally better on this KPI, as they achieved roughly the same slight performance in OR. The proposal significantly reduces CBR by a percentage varying from 10% to 20%.

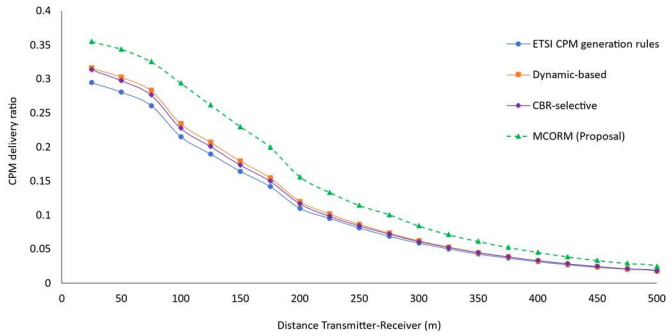

Fig. 6. The CPM delivery ratio as a function of the distance transmitter-receiver

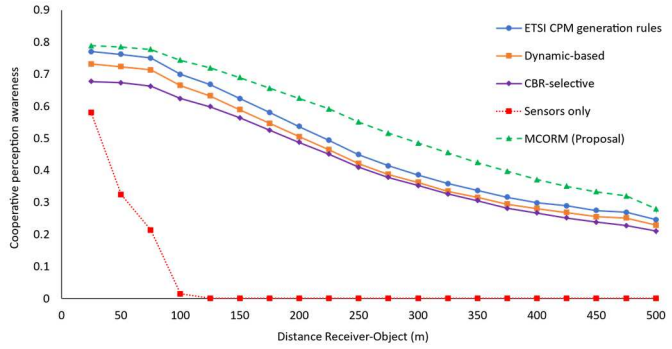

Fig. 7. The CPA as a function of the distance between the perceived object and the CAV receiving it. In the case of "Sensors only", the X-axis represents the distance from the object to the CAV detected it using its onboard sensors.

Reducing CBR improves the reliability of V2V communication. We measure this reliability using CDR plotted in Fig. 6. The figure plots CDR variation in the V2V network for each approach as a function of the distance between the transmitter and the receiver CAVs. As observed, CDR results in poor values at short, medium, and long distances as it is mostly affected by the propagation conditions due to the presence of buildings in the driving scenario. Fig. 6 shows that the ETSI CPM generation rules result in lower probabilities for short distances of less than 100 m because it has attained the highest levels in OR and CBR. However, the dynamic-based technique and the CBR-selective scheme improve slightly on this metric at the same distances. The proposal considerably increases CDR over all distances thanks to the improvements achieved in CBR.

Typically, the increase in CDR enables CAVs to receive additional useful objects via CPMs, resulting in increased CPA in the V2V network. This is depicted in Fig. 7, which compares the CPA levels reached by each approach. The figure also depicts the CPA level attained using only onboard sensors without V2V communication. Fig. 7 indicates that relying exclusively on the onboard sensors results in a poor perception of the driving environment. However, this limitation is overcome by including the exchange of CP information between CAVs. Compared to the ETSI CPM generation rules, the dynamic-based technique and the CBR-based scheme have almost attained the same CPA at distances larger than 100 m; however, they degrade this metric by around 5% and 10%, respectively, at distances smaller than 10 m.

On the other hand, the proposal improves CPA at distances less than 100 m which are critical for the safety of CAVs. Moreover, Fig. 7 shows that the proposal increases CPA at medium and long distances from 100 to 400 m. This is expressed by the performances achieved by CBR and CDR, which have enabled CAVs to receive more CPM objects that seem to be lost or not sent using the other approaches.

VI. Conclusion

This paper introduced a multi-agent advantage actor-critic learning method in a multi-CAV driving environment. Its primary objective is to learn each CAV a CPM content selection policy that maximizes object usefulness for the receiving CAVs to mitigate redundancy in the V2V network. The proposal is evaluated by simulation and compared to the state-of-the-art approaches based on various KPIs. Results showed the ability of CAVs to learn CPM content selection policies while maximizing usefulness efficiently. Results also demonstrated that the proposal considerably reduced redundant objects without static thresholds while maintaining cooperative perception awareness in the V2V network. In our future work, we will study the performances of the proposed method by including the exchange of CP information with roadside infrastructure using V2I communications.

References

[1] I. Soto, M. Calderón, O. Amador, U. Urueña, "A survey on road safety and traffic efficiency vehicular applications based on C-V2X technologies," Vehicular Communications 2O22, 33, 100428. https://doi.org/10.1016/j.vehcom.2021.100428.

[2] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-everything (v2x) services supported by LTE-based systems and 5G," IEEE Commun. Standards Mag., vol. 1, no. 2, 2017, pp. 70–76, https://doi.org/10.1109/MCOMSTD.2017.1700015.

[3] ETSI EN 302 663 V1.3.11, Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency bands, 2020.

[4] ETSI EN 302 637-2 V1.3.2, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service", 2014.

[5] ETSI TR 103 562-V2.1.1, Intelligent Transport System (ITS); Vehicular Communications. Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2, 2019.

[6] K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," in IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26-38, Nov. 2017, https://doi.org/doi: 10.1109/MSP.2017.2743240.

[7] VP. Rekkas, S. Sotiroudis, P. Sarigiannidis, S Wan, GK. Karagiannidis, SK. Goudos, "Machine Learning in Beyond 5G/6G Networks—State-of-the-Art and Future Trends," Electronics. 2021, 10(22):2786. https://doi.org/10.3390/electronics10222786.

[8] V. R. Konda and J. N. Tsitsiklis, “Actor-critic algorithms,” in Proceedings of the Conference on Neural Information Processing Systems, 2000, pp. 1008–1014.

[9] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal difference learning with function approximation,” IEEE Trans. Autom. Control, vol. 42, no. 5, 1997, pp. 674–690.

[10] V. Mnih, et al., “Asynchronous methods for deep reinforcement learning,” International conference on machine learning, 2016, pp. 1928–1937, https://doi.org/10.48550/arXiv.1602.01783.

[11] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in Advances in neural information processing systems, 2017, pp. 6379–6390, https://doi.org/10.48550/arXiv.1706.02275.

[12] F. A. Oliehoek, C. Amato, et al., “A concise introduction to decentralized POMDPs,” vol. 1. Springer, 2016.

[13] R. Riebl, H. Günther, C. Facchi and L. Wolf, “Artery: Extending Veins for VANET applications,” 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015, pp. 450-456, https://doi.org/10.1109/MTITS.2015.7223293.

[14] Krajzewicz, Daniel & Erdmann, Jakob & Behrisch, “Michael & Bieker-Walz, Laura. (2012),” Recent Development and Applications of SUMO - Simulation of Urban Mobility,” International Journal On Advances in Systems and Measusrements. 3&4.

[15] G. Thandavarayan, M. Sepulcre and J. Gozalvez, “Analysis of Message Generation Rules for Collective Perception in Connected and Automated Driving,” 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 134-139, https://doi.org/10.1109/IVS.2019.8813806.

[16] G. Thandavarayan, M. Sepulcre and J. Gozalvez, “Redundancy Mitigation in Cooperative Perception for Connected and Automated Vehicles,” 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1-5, https://doi.org/10.1109/VTC2020-Spring48590.2020.9129445 .

[17] A. Chtourou, P. Merdrignac, O. Shagdar, “Context-aware content selection and message generation for collective perception services,” Electronics, vol. 10, no. 20, 2021, https://doi.org/10.3390/electronics10202509.

[18] H. Huang, W. Fang and H. Li, “Performance Modelling of V2V based Collective Perceptions in Connected and Autonomous Vehicles,” 2019 IEEE 44th Conference on Local Computer Networks (LCN), 2019, pp. 356-363, https://doi.org/10.1109/LCN44214.2019.8990854.

[19] H. Huang, H. Li, C. Shao, T. Sun, W. Fang and S. Dang, “Data Redundancy Mitigation in V2X Based Collective Perceptions,” in IEEE Access, vol. 8, pp. 13405-13418, 2020, https://doi.org/10.1109/ACCESS.2020.2965552.

[20] B. Jung, J. Kim and S. Pack, “Deep Reinforcement Learning-based Context-Aware Redundancy Mitigation for Vehicular Collective Perception Services,” 2022 International Conference on Information Networking (ICOIN), 2022, pp. 276-279, https://doi.org/10.1109/ICOIN53446.2022.9687254.