Explainable AI-aided Feature Selection and Model Reduction for DRL-based V2X Resource Allocation

Nasir Khan, *Graduate Student Member, IEEE*, Asmaa Abdallah, *Member, IEEE*, Abdulkadir Celik, *Senior Member, IEEE*, Ahmed M. Eltawil, *Senior Member, IEEE*, and Sinem Coleri, *Fellow, IEEE*

***Abstract*—Artificial intelligence (AI) is expected to significantly enhance radio resource management (RRM) in sixth-generation (6G) networks. However, the lack of explainability in complex deep learning (DL) models poses a challenge for practical implementation. This paper proposes a novel explainable AI (XAI)-based framework for feature selection and model complexity reduction in a model-agnostic manner. Applied to a multi-agent deep reinforcement learning (MADRL) setting, our approach addresses the joint sub-band assignment and power allocation problem in cellular vehicle-to-everything (V2X) communications. We propose a novel two-stage systematic explainability framework leveraging feature relevance-oriented XAI to simplify the DRL agents. While the former stage generates a state feature importance ranking of the trained models using Shapley additive explanations (SHAP)-based importance scores, the latter stage exploits these importance-based rankings to simplify the state space of the agents by removing the least important features from the model's input. Simulation results demonstrate that the XAI-assisted methodology achieves ∼97% of the original MADRL sum-rate performance while reducing optimal state features by ∼28%, average training time by ∼11%, and trainable weight parameters by ∼46% in a network with eight vehicular pairs.**

***Index Terms*—Explainable AI (XAI), deep reinforcement learning (DRL), vehicle-to-everything (V2X) communications, short-packet transmission, ultra-reliable and low-latency communications (URLLC).**

I. Introduction

CELLULAR vehicle-to-everything (C-V2X) networks have emerged as key enablers for intelligent transportation systems, leveraging existing cellular infrastructure to improve road safety, traffic efficiency, and in-vehicle entertainment. C-V2X technology promises widespread coverage, high reliability, and efficient spectrum use, even in high-mobility scenarios [1]. Advanced C-V2X applications demand transmission latencies of a few milliseconds and 99.999% reliability [2]. As vehicular networks expand and radio resource management becomes increasingly complex, existing centralized resource allocation approaches in cellular networks struggle to meet diverse quality-of-service (QoS) requirements, particularly ultra-reliable and low-latency needs.

Nasir Khan and Sinem Coleri are with the department of Electrical and Electronics Engineering, Koc University, Istanbul, Turkey, email: {nkhan20, scoleri}@ku.edu.tr. This work is supported by Scientific and Technological Research Council of Turkey Grant #119C058 and Ford Otosan.

Asmaa Abdallah, Abdulkadir Celik, and Ahmed M. Eltawil are with the Computer, Electrical, and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia (e-mail: asmaa.abdallah@kaust.edu.sa; abdulkadir.celik@kaust.edu.sa; ahmed.eltawil@kaust.edu.sa)

The recent decade has witnessed a surge in publications unveiling the merits of artificial intelligence (AI) and deep learning (DL) methods, suggesting their potential to mitigate the aforementioned challenges of model-based approaches [3]. In particular, deep reinforcement learning (DRL) has been shown effective in solving hard-to-optimize non-convex combinatorial problems efficiently and has widely been adopted to deal with high-dimensional state-action spaces of vehicular communications applications [4]. Nevertheless, DRL-based resource allocation relies on complex deep neural networks (DNNs) with multiple layers, making them difficult to understand due to their black-box nature and lack of explainability [5]. This complexity obscures the influence of input state features on the model's output and the logical reasoning behind decisions. This lack of explainability hinders the adoption of DRL-based resource allocation in safety-critical vehicular applications, as the decision-making processes are not easily understandable.

A. Relevant works

Resource sharing in C-V2X applications requires a judicious allocation for mitigating interference and optimizing resource utilization for efficient vehicular communications. In centralized resource management in C-V2X, the BS collects the channel statistics to monitor the quality of each link and formulate the optimal resource allocation objective function [6]–[9]. For instance, centralized graph theory-based proposals [6] and hierarchical optimization theory-based solutions [7] require executing iterative algorithms at the central controller with up-to-date global channel state information (CSI) availability. To alleviate the CSI acquisition overhead, low-complexity algorithms based on only large-scale fading information [8] or imperfect CSI assumptions [9] have been proposed. However, the V2X resource allocation problem is often modeled as a combinatorial optimization problem with nonlinear constraints, which are difficult to solve using traditional optimization algorithms in real time. Further, these works [6]–[9] consider Shannon rate-based communication, ignoring the reliability aspect of V2V links in terms of the decoding error probability, implicitly assuming infinite block-length availability, making them unsuitable for the low-latency, high-reliability requirements of V2X scenarios addressed in this study.

Many existing studies have considered different DRL-based solutions for solving resource allocation problems in V2X

communication networks [10]–[13]. Progress has also been made to integrate DRL with federated learning [14], [15], meta-learning [16], graph theory [17] and digital twin-driven V2X networks [18] for improved performance and fast adaptability of the resource allocation policy in the dynamic V2X environments. However, to the best of the authors' knowledge, none have addressed the open problem of providing explainability to these solutions. These aforementioned studies [10]–[18] commonly focus on the performance and robustness of the underlying DRL-based strategies and can yield a good solution for V2X resource allocation problems. However, they do not provide insights into the workings of these complex model-based solutions and, as such, suffer from a performance-explainability trade-off.

To strike a good balance between performance and explainability trade-off, explainable AI (XAI) is gaining significant momentum in wireless communications to aid the understanding of complex AI models [19]. XAI encompasses processes and methods that reveal the inner workings of complex AI models, helping to balance the trade-off between explainability and performance inherent to these models. The Shapley additive explanations (SHAP) method is particularly effective for feature selection, as it assigns importance values to each feature based on their contribution to the model's predictions, allowing for the exclusion of less significant features without sacrificing performance [20]. SHAP has a strong theoretical foundation based on Shapley values in game theory and exhibits several desirable properties compared to other XAI methods. First, SHAP solutions satisfy three essential properties for any additive feature attribution model: local accuracy, missingness, and consistency [21]. Second, SHAP combines the local interpretable model-agnostic explanations (LIME) [22] method, which explains individual predictions using an interpretable model, with Shapley values. This integration allows for faster computation of explanations for complex learning models than directly calculating Shapley values.

Within the wireless communications literature, research on the XAI application remains limited. Several studies have begun integrating XAI techniques to enhance transparency and understanding within complex network systems. For instance, classical XAI methods and SHAP have been recommended to analyze the root cause of Service Level Agreement (SLA) violation prediction in a 5G network slicing setup [23]. KernelSHAP is invoked to quantify the importance of input feature contributions to the model's outcome and demystify model behavior for short-term resource reservation (STRR) in network slicing operations [24]. A two-stage pipeline is introduced for robust network intrusion detection using extreme gradient boosting (XGBoost), and SHAP is employed to explain intrusion attacks [25]. Further, XAI is integrated with federated learning to predict and interpret network slices' latency key performance indicator (KPI) [26]. XAI is utilized to design a composite reward mechanism assisted by SHAP importance scores to encourage DRL agents to learn the best actions for specific network slices in SLA-aware 6G network slicing setup [27]. A trust-aware, explainable federated DRL model is developed using SHAP values to compute the trust score for trajectory and velocity decision-making in autonomous driving. The trust score is used to select and aggregate vehicles participating in the federated learning process [28]. The scope of the above works [24]–[28] does not extend beyond providing insights into the feature relevance in the decision-making process of the underlying machine learning models in different wireless settings. Additionally, the XAI metrics (trust score, XRL reward) in [27], [28] are tied to the underlying algorithm and cannot be applied directly to other learning algorithms without modification.

XAI-based methods for feature selection have garnered significant attention in wireless communications to determine the primary attributes influencing the model decision and enhancing model performance. Feature selection aims to identify a minimal set of features that optimizes the objective function without significantly degrading performance. Traditional filter-based feature selection methods assess importance based on data characteristics but overlook algorithm dependence and feature interactions. Wrapper methods (e.g., Recursive Feature Elimination) improve accuracy but are computationally expensive for large datasets. XAI-based feature selection methods are generally architecture-specific, data-dependent, and provide local explanations [22]. In contrast, SHAP stands out as a popular and stable XAI method owing to its robust mathematical foundation embedded in the game theory concept of Shapley values. It provides a fair allocation of rewards among inputs, considering their individual/collective contributions and providing both local/global explanations.

To our knowledge, no prior work has provided a systematic approach for generating feature importance explanations for complex AI models in multi-agent deep reinforcement learning (MADRL) settings or applied these explanations to enhance understanding or reduce the execution time and model complexity (e.g., via feature selection). However, similar methods have been used for feature importance in other networking areas. For instance, feature importance scoring for tree-based classifiers in wireless intrusion detection using the SHAP XAI method is derived in [29]. By leveraging these importance scores, the AI classifiers are simplified, resulting in a 90% reduction in model size without any significant performance loss. XAI-based feature selection methodology comparing counterfactual and rule-based explanation scores with SHAP scores for feature ranking is used to simplify trained models in energy-efficient resource management in 5G networks [30].

B. Contributions

In this work, we present a distributed multi-agent deep reinforcement learning (MADRL) algorithm to jointly optimize transmit power and spectrum sub-band assignment, enhancing the performance of vehicle-to-network (V2N) and vehicle-to-vehicle (V2V) links in ultra-reliable and low-latency communications (URLLC)-enabled V2X communications network. Utilizing the SHAP method from XAI, we design a post-hoc and model-agnostic explainability pipeline that improves understanding of DRL agent inferences and simplifies the agent's inputs through feature selection based on importance

rankings.[1] The contributions of this paper are summarized as follows:

- We formulate the joint optimization problem of spectrum and transmit power allocation with the objective of maximizing the sum-rate of V2N and V2V links under reliability, latency, and transmit power constraints in a URLLC-enabled vehicular network. We propose a MADRL algorithm with centralized learning and decentralized execution to solve the formulated optimization problem. Specifically, multiple DQNs are distributively executed at the V2V transmitters, whereas the weight parameters are centrally trained at the base station (BS) using a common reward function for all agents to ease implementation and improve stability.
- We propose a novel two-stage systematic explainability framework leveraging feature relevance-oriented XAI to simplify the DRL agents. The former stage involves generating state feature importance ranking of the trained models using the SHAP-based importance scores. The latter stage exploits these importance-based rankings to simplify the state space of the agents, whereby the least important features are removed from the model's input, i.e., by masking feature values and observing the effect on the model's performance using an automated novel variable feature selection method.
- We quantitatively validate the effectiveness of the proposed XAI-based methodology. Via extensive simulations, we demonstrate that our proposed methodology can simplify the model in terms of the average training time, the number of broadcast parameters, and the number of optimal state features required for training for different network configurations without significant performance loss.

C. Notations and Paper Organization

The boldface lowercase letter is used to represent a column vector. $\leftarrow$ denotes the assignment operation. $\|\mathbf{x}\|$ denotes the length of $\mathbf{x}$. $|\cdot|$ is the absolute function operator and $f \equiv g$ denotes that function f is equivalent to g. Key mathematical notations are summarized in Table I.

The rest of the paper is organized as follows. Section II describes the vehicular URLLC system model and assumptions used in the paper. Section II-B presents the mathematical formulation of the joint transmit power and spectrum sub-band allocation in vehicular URLLC system. Section III provides a brief background on the design of the DRL framework and describes the proposed multi-agent DRL-based algorithm for the transmit power and spectrum sub-band allocation problem. Section IV explains the proposed systematic SHAP-based XAI methodology for generating the feature importance ranking and describes the proposed feature selection algorithm.

[1]A preliminary version of this work is described as a case study in [5], where we devise a fixed threshold-based input feature selection strategy utilizing the SHAP-based feature importance ranking in a power control problem in V2X networks. Different from our previous work [5], we extend the framework to include sub-band allocation and propose a novel post-hoc explainability framework with an automated variable state feature selection strategy.

Table I: Table of notations used in this work

Notation	Stands For
$\mathcal{N}, \mathcal{K}$	Set of V2N links and V2V links
$g_k^{(t)}[n]$	Direct channel gain of the k-th V2V link
$g_{k'\to k}^{(t)}[n]$	Interfering channel gain from k'-th V2V link to k-th V2V receiver
$g_{k\to B}^{(t)}[n]$	Interfering channel gain from k-th V2V link to to BS
$g_{n\to B}^{(t)}[n]$	Direct channel gain from n-th V2N link to the BS
$g_{n\to k}^{(t)}[n]$	Interfering channel gain from n-th V2N link to k-th V2V receiver
$P_n^{V2N\,(t)}$	Transmit power of the n-th V2N link
$P_k^{V2V\,(t)}[n]$	Transmit power of the k-th V2V link
$\eta_k^{(t)}[n]$	Resource allocation indicator of k-th V2V pair
$\varepsilon_k^{(t)}[n]$	decoding error probability at the k-th V2V receiver
$\epsilon_{\max,k}$	Maximum error probability
$\mathcal{D}, D$	Replay buffer and mini-batch size
$\mathcal{S}_k$	SHAP matrix for k-th agent
$\mathcal{X}_{BG}, \mathcal{X}$	Background and hold-out datasets
$\mathbf{x}$	Input state feature vector
$f_k(\mathbf{x})$	Prediction model for k-th agent
$\boldsymbol{\phi}_k(\mathbf{x})$	SHAP_values for the k-th agent
$\tilde{\boldsymbol{\Phi}}_{k,l}$	Transformed SHAP_values for the k-th agent
$\widehat{\boldsymbol{\Phi}}$	Average $\tilde{\boldsymbol{\Phi}}_{k,l}$ across the K agents
L, M	Total number of state features / valid actions
Δ	Precision threshold
α	Average network performance metric

Section V evaluates the performance of the proposed solution strategy. Finally, conclusions and future research directions are presented in Section VI.

II. System Model and Problem Formulation

This section first provides the system model of the C-V2X communications network. Then, it formulates a resource allocation problem that aims to maximize the sum-rate for the V2N links as well as to ensure reliable information transmission for the V2V links.

A. System Model

We consider a single-cell C-V2X communications network covered by a Road-Side Unit (RSU) acting as a single-antenna BS, as shown in Fig. 1. Each vehicle is equipped with a single-antenna On-Board Unit (OBU), supporting high data rate uplink services and reliable safety message sharing. C-V2X includes two operation modes to support diverse vehicular applications: V2N and V2V mode. V2N links support high data rate applications (e.g., video streaming, location tracking, map updates) via cellular interfaces using the Uu interface of 5G new radio (NR) [2]. V2V links transmit safety-related information through device-to-device (D2D) communications, requiring low latency and high reliability. We represent the sets of V2N and V2V links as $\mathcal{N}$ and $\mathcal{K}$, with cardinalities N and K, respectively. Each V2N link occupies a unique sub-band with fixed transmission power. In cases where $N > K$, we introduce $N - K$ virtual V2V connections. Our focus is on Mode 4 in the cellular V2X architecture, where vehicles autonomously select radio resources for V2V communications

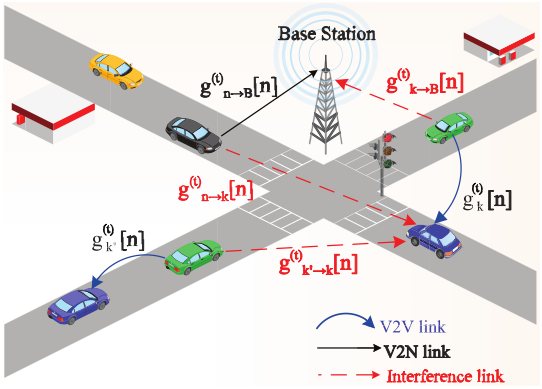


Figure 1: System model for the V2X communication network.

[31]. Each V2V link can reuse an orthogonal sub-band of the V2N links to improve spectrum utilization, with multiple V2V links potentially reusing the same sub-band, which may cause interference.

1) Channel Models: Consider a slotted communication system with scheduling time slots indexed by t, where each slot has a duration of τ. We consider a quasi-static block fading channel model where orthogonal frequency division multiplexing (OFDM) technology is utilized to group consecutive subcarriers to form a spectrum sub-band. The channel coherence time is given by $T_c = \sqrt{\frac{9}{16\pi f_D^2}}$, where $f_D = \frac{f_c \cdot v_s}{c}$ is the Doppler frequency, f_c is the carrier frequency, v_s is the vehicle velocity, and c is the speed of light [32]. $T_c > \tau$ holds for moderate vehicular speeds, and the channel state information (CSI) can be regarded as constant throughout the slot duration. Then, the direct channel gain of the k-th V2V link over the n-th sub-band in coherence time slot t is expressed as $g_k^{(t)}[n] = \alpha_k^{(t)} \left|h_k^{(t)}[n]\right|^2$, where $h_k^{(t)}[n] \sim \mathcal{CN}(0,1)$ is a complex Gaussian variable representing the Rayleigh fading, and $\alpha_k^{(t)}$ captures the large-scale fading effect, including path loss and shadowing, which is typically frequency-independent and changes slowly since it primarily depends on the locations of the transmitter and receiver of the V2V pair. The channel gain between the n-th V2N transmitter and the BS over the n-th sub-band at slot t is defined as $g_{n\to B}^{(t)}[n] = \alpha_{n\to B}^{(t)} \left|h_{n\to B}^{(t)}[n]\right|^2$, where subscript $n \to B$ denotes the n-th V2N link traversing the BS.

In the case of n-th sub-band's reuse, the interfering channel gains between the i-th and the j-th V2V/V2N links are defined as $g_{i\to j}^{(t)}[n] = \alpha_{i\to j}^{(t)} \left|h_{i\to j}^{(t)}[n]\right|^2$, where $i \neq j$, $i \in \{k, n\}$, $j \in \{k, B\}$, $\alpha_{i\to j}^{(t)}$ represents the large-scale fading, and $h_{i\to j}^{(t)}[n]$ denotes the small-scale fading.

2) Achievable Rates: The achievable data rate of the n-th V2N link at time slot t is expressed as

$$R_n^{V2N\,(t)}[n] = B_w \log_2\left(1 + \gamma_n^{V2N\,(t)}[n]\right), \tag{1}$$

where B_w indicates the channel bandwidth occupied by each V2N link and $\gamma_n^{V2N\,(t)}[n]$ is the received signal-to-interference-plus-noise ratio (SINR) of the n-th V2N link over the n-th sub-band at time slot t, which is defined as

$$\gamma_n^{V2N\,(t)}[n] = \frac{P_n^{V2N\,(t)} g_{n\to B}^{(t)}[n]}{\sigma^2 + \sum_{k\in\mathcal{K}} \eta_k^{(t)}[n] P_k^{V2V\,(t)}[n] g_{k\to B}^{(t)}[n]}, \tag{2}$$

where $P_n^{V2N\,(t)}$ and $P_k^{V2V\,(t)}[n]$ denote the transmit powers of the n-th V2N link and the k-th V2V link over the n-th sub-band, respectively; σ^2 represents the variance of additive white Gaussian noise; and $\eta_k^{(t)}[n] \in \{0, 1\}$ is the resource allocation indicator with $\eta_k^{(t)}[n] = 1$ implying the k-th V2V link uses the spectrum of the n-th V2N link and $\eta_k^{(t)}[n] = 0$ otherwise. We assume that each V2V link can only use one sub-band at the same time, i.e., $\sum_{n\in\mathcal{N}} \eta_k^{(t)}[n] = 1$, while the orthogonal sub-band of each V2N link can be reused by multiple V2V pairs.

URLLC-enabled V2V communications are mainly responsible for the reliable dissemination of safety-critical messages using short code block lengths, rendering the classic Shannon capacity no longer appropriate to describe the maximum achievable data rate. In the context of finite block length regime, the maximum achievable information rate of the k-th V2V link at time slot t over the n-th sub-band, which can be decoded with block error probability no greater than ε_k is given by the normal approximation [33]

$$R_k^{V2V\,(t)}[n] = B_w \left(C\left(\gamma_k^{V2V\,(t)}[n]\right) - \sqrt{\frac{V_k^{(t)}[n]}{m}} \frac{f_{Q^{-1}}(\varepsilon_k^{(t)}[n])}{\ln 2} \right), \tag{3}$$

where $C\left(\gamma_k^{V2V\,(t)}[n]\right) = log_2\left(1 + \gamma_k^{V2V\,(t)}[n]\right)$ is the Shannon rate, $V_k^{(t)}[n] \triangleq 1 - \left(1 + \gamma_k^{V2V\,(t)}[n]\right)^{-2}$ is the channel dispersion, m is the blocklength allocated to the k-th V2V link, $\varepsilon_k^{(t)}[n] \in (0, 1)$ is the decoding error probability at the k-th V2V receiver over n-th sub-band, $f_{Q^{-1}}(\cdot)$ is the inverse Gaussian tail function $f_Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$, and $\gamma_k^{V2V\,(t)}[n]$ is the SINR of the k-th V2V link over the n-th sub-band at time slot t defined as

$$\gamma_k^{V2V\,(t)}[n] = \frac{P_k^{V2V\,(t)}[n] g_k^{(t)}[n]}{\sigma^2 + I_k^{(t)}[n]}, \tag{4}$$

where $I_k^{(t)}[n] = P_n^{V2N\,(t)} g_{n\to k}^{(t)}[n] + \sum_{k'\neq k} \eta_{k'}^{(t)}[n] P_{k'}^{V2V\,(t)}[n] g_{k'\to k}^{(t)}[n]$ is the collective interference at the k-th V2V receiver from the n-th V2N link and the k'-th V2V links sharing the n-th sub-band. The code block length can be expressed by $m = B_w \Delta_T$, where Δ_T represents the packet transmission latency defined as the time taken for an entire packet of L bits to be transmitted. In our work, packet transmission latency is fixed to 1 milliseconds to ensure the latency requirement defined in 3GPP TR 36.885 [34]. Then, for a given payload size of L bits per vehicle, the coding/data rate for the k-th V2V link, $\tilde{r} = \frac{L}{B_w \Delta_T}$. By rearranging Eq. (3), the decoding error probability at the k-th V2V receiver can be expressed as

$$\varepsilon_k^{(t)}[n] \approx f_Q\left(\ln 2 \sqrt{\frac{B_w \Delta_T}{V_k^{(t)}[n]}} \left(C\left(\gamma_k^{V2V\,(t)}[n]\right) - \frac{L}{B_w \Delta_T} \right) \right). \tag{5}$$

Then, based on the joint coding theory [35], the achievable

rate of the k-th V2V link can be expressed as

$$\mathcal{T}_k^{(t)} = \sum_{n=1}^{N} \eta_k^{(t)}[n] \left(R_k^{V2V\,(t)}[n] \left(1 - \varepsilon_k^{(t)}[n]\right)\right). \tag{6}$$

B. Problem Formulation

To maximize the sum-rate for the V2N and V2V links while ensuring reliable information transmission for the V2V links, we jointly optimize the spectrum allocation $\eta_k^{(t)}[n]$ and the V2V transmission power $P_k^{V2V\,(t)}[n]$, $\forall k \in \mathcal{K}$ and $\forall n \in \mathcal{N}$. Specifically, the sum-rate performance metric measures the total data rate achieved by all V2V and V2N communication links in the network. This metric is crucial in V2X communications because it directly reflects the network's ability to efficiently utilize the available spectrum while supporting multiple simultaneous links. A higher sum-rate indicates better spectral efficiency for data-intensive applications and reliable transmission of safety-critical messages. Considering these QoS requirements in vehicular networks, the objective function of our work is to jointly optimize the sum rate for the V2N links and the V2V links. The dynamic multi-objective resource allocation problem at time slot t is formulated as

$$\underset{\boldsymbol{\eta}^{(t)}, \boldsymbol{P}^{(t)}}{\text{maximize}} \left\{ \omega_1 \sum_{n=1}^{N} R_n^{V2N\,(t)} + \omega_2 \sum_{k=1}^{K} \mathcal{T}_k^{(t)} \right\} \tag{7a}$$

subject to

$$0 \le \epsilon_k^{(t)}[n] \le \varepsilon_{\max,k} \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{7b}$$

$$0 \le P_k^{V2V\,(t)}[n] \le P_{\max}, \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{7c}$$

$$\sum_{n \in \mathcal{N}} \eta_k^{(t)}[n] = 1, \quad \forall k \in \mathcal{K} \tag{7d}$$

$$\eta_k^{(t)}[n] \in \{0, 1\}. \tag{7e}$$

where ω_1 and ω_2 represent the weights to balance the sum-rate of V2N and V2V link, respectively; and $\boldsymbol{\eta}^{(t)} = \left[\eta_1[1], \ldots, \eta_k[n], \ldots, \eta_K[N]\right]$, $\mathbf{P}^{(t)} = \left[P_1^{V2V}[1], \ldots, P_k^{V2V}[n], \ldots, P_K^{V2V}[N]\right]$ are the vectors for the sub-band selection and the power allocations at time slot t, respectively. Then, in (7b), $\epsilon_{\max,k}$ is the maximum error probability constraint for the k-th V2V link to ensure a predefined quality of service (QoS). (7c) is the maximum power constraint for the k-th V2V transmitter. (7d) indicates that each V2V link can only use one sub-band at a time.

The optimization problem (7) is a non-convex mixed integer nonlinear programming (MINLP) problem and involves sequential decision-making over the time slots. Due to the dynamic nature of the C-V2X networks and the associated short coherence time interval, centralized solutions are inadequate due to difficulties in acquiring the instantaneous CSI of all links [36]. To address these issues, we propose to exploit DRL to solve the problem (7) in a decentralized fashion. Specifically, the sequential decision-making over time can be encapsulated within a Markov decision process (MDP) framework, and solved using a DRL-based strategy by treating each V2V transmitter as an agent interacting with the unknown environment to maximize the reward. In what follows, we transform the problem of joint spectrum and power allocation into a multi-agent DRL problem.

III. Multi-agent Reinforcement Learning-based Solution

In this section, we design a DRL-based algorithm to solve the resource allocation problem in (7). To build the foundations for the proposed DRL-based strategy, the description of key components of the RL framework in terms of the state space, action space, and reward function is first provided, followed by the proposed MADRL algorithm.

A. Problem Transformation into DRL Framework

DRL effectively addresses complex decision-making problems by training an agent to interact with its environment and learn optimal behaviors over time through continuous feedback and adjustment. The agent refines its actions to achieve optimal outcomes within a MDP framework [37]. An MDP is encapsulated by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is the transition probability from states to actions, and $\mathcal{R}$ is the reward for taking action in a state. At each discrete time step t, the agent observes the state $s^{(t)}$, takes action $a^{(t)}$ according to policy $\pi(s^{(t)}, a^{(t)})$, transitions to state $s^{(t+1)}$, and receives a reward $r^{(t)}$. The DRL-based system design aims to learn an optimal policy that maximizes the expected return [38], [39]. Therefore, we cast the optimization problem (7) as an MDP by mapping the key elements from the DRL framework to the resource allocation problem.

In the multi-agent scenario, each V2V transmitter acts as an agent and interacts with the unknown communication environment. The joint action $\bar{\mathcal{A}}^{(t)} = \left(a_1^{(t)}, \cdots, a_k^{(t)}, \cdots a_K^{(t)}\right)$, with $a_k^{(t)}$ as the k-th V2V agent's action in the time slot t, collectively influences the common environment and directs the agents towards the optimal policy. Hence, in our MADRL framework, different agents compete for limited power and spectrum resources, and the resource-sharing problem is shifted to a fully cooperative one by providing all agents with the same common reward. Next, we introduce the key components of the DRL framework in detail in terms of the state space, action space, and reward function.

*1) **State Space**:* The state of the k-th V2V agent at time slot t, $s_k{}^{(t)}$, includes the channel information and the received interference from other links. Specifically, the channel information for the k-th V2V agent includes the instantaneous channel gain of its own link $g_k^{(t)}[n]$, $\forall n \in \mathcal{N}$, the interference channel gain from the transmitter of the n-th V2N link and the k'-th V2V link, $g_{n\to k}^{(t)}[n]$ and $g_{k'\to k}^{(t)}[n]$, $\forall n \in \mathcal{N}$, $(k' \neq k)$, and the interference channel gain from its transmitter to the BS $g_{k\to B}^{(t)}[n]$, $\forall n \in \mathcal{N}$. The channel information $g_k^{(t)}[n]$, $g_{n\to k}^{(t)}[n]$, and $g_{k'\to k}^{(t)}[n]$, can be accurately estimated by the receiver of the k-th V2V link at the beginning of each scheduling slot, and we assume it is also available instantaneously at the transmitter through delay-free feedback. The CSI for $g_{k\to B}^{(t)}[n]$ is estimated at the BS and broadcast to all vehicles within the BS coverage at the beginning of each scheduling slot [11], [38].

Additionally, the aggregate interference from the V2N links and the V2V links over the n-th sub-band in the past time slot

$I_k^{(t-1)}[n]$ is included in the state space of the k-th agent, which is expressed as

$$I_k^{(t-1)}[n] = P_n^{V2N(t-1)} g_{n\to k}^{(t)}[n] + \sum_{k'\neq k} \eta_{k'}[n] P_{k'}^{V2V(t-1)}[n] g_{k'\to k}^{(t)}[n],$$

where the first term is the interference from the V2N transmitter and the second term is the aggregate interference from all V2V transmitters using the same n-th sub-band in the previous time slot. It should be noted that the sum interference for the k-th V2V agent is estimated in slot t as follows. At the beginning of the current time slot, the new transmit power and spectrum allocation decision of each V2V agent has not been determined and each V2V agent utilizes the resource decision of the previous time slot to calculate $I_k^{(t-1)}[n]$, although the CSI of the whole network has changed. Then, the current state $s_k^{(t)}$ of the k-th V2V agent is given by

$$s_k^{(t)} = \left\{ \left(g_k^{(t)}[n], g_{k'\to k}^{(t)}[n], g_{k\to B}^{(t)}[n], g_{n\to k}^{(t)}[n], I_k^{(t-1)}[n] \right) \mid n \in N \right\}. \tag{8}$$

The scale of the state space is $(K+2)\times N$ per V2V agent, which includes N elements for the direct link between the k-th V2V transmitter and its receiver, $(K\times N - N)$ elements for the interfering channels from other V2V transmitters, N elements for the V2N links, and N elements for the sum interference.

*2) **Action Space**:* The action of each agent at time slot t corresponds to the spectrum sub-band and transmission power selection. While N disjoint sub-bands are preoccupied by the N V2N links and all V2V links share these sub-bands, each V2V agent can only pick at most one spectrum sub-band during the same time slot $\left(\sum_{n\in\mathcal{N}} \eta_k^{(t)}[n] = 1, \forall \mathcal{K}\right)$. The power allocation is discretized in L_Q levels given by $\mathcal{P} = \left\{\frac{P_{\max}}{L_Q}, \frac{2P_{\max}}{L_Q}, \cdots, P_{\max}\right\}$, with $\mathcal{P}$ representing the power action set. Then, the action executed by the k-th V2V agent in the time slot t can be expressed as $a_k^{(t)} = \left[\eta_k^{(t)}[n], P_k^{V2V(t)}[n]\right]$, which includes $\eta_k^{(t)}[n]$ and $P_k^{V2V(t)}[n]$ for spectrum sub-band and power action decision, respectively. Accordingly, each agent has an action vector with the dimension of $L_Q \times N$.

*3) **Reward Function**:* A common reward is designed for all agents to encourage cooperation among the multiple V2V agents. In order to reflect the performance of the decision taken by the agents, the immediate reward function is formulated as

$$\begin{aligned} r^{(t)} =& \lambda_1 \sum_{n=1}^{N} R_n^{(t)\,V2N} + \lambda_2 \sum_{k=1}^{K} \mathcal{T}_k^{(t)} \\ &- \lambda_3 \sum_{k=1}^{K} \left(\max\left(\epsilon_k^{(t)} - \epsilon_{max,k}, 0 \right) \right), \end{aligned} \tag{9}$$

where the summation terms respectively represent the sum-rate of V2N links, the sum-rate of V2V links, and the reliability requirements of the V2V links; and λ_i, $i \in \{1,2,3\}$, are the positive weights to balance the utility and the penalty cost in terms of constraint violations. To align the designed reward function at each step with the network objective (7a), the first two terms in (9) are included as positive reward elements. Additionally, the reward function incorporates a penalty for failing to satisfy the reliability constraint related to the decoding error probability. The selection of weight coefficients is critical, as they significantly affect the learning efficiency and convergence of the algorithm. In practical scenarios, these coefficients, denoted as λ_i, are considered hyperparameters and require empirical tuning. Setting equal positive weights for all reward components prioritizes maximizing the sum rate of V2V and V2N links. However, our tuning experience suggests such a design will impede the algorithm's convergence as the agents struggle to learn reliability aspects. This is because the penalty term, based on target error probabilities, is much smaller than other reward components. The reward function components are normalized by setting $\lambda_1 = \frac{1}{5K}$, $\lambda_2 = \frac{1}{R'}$, where $R' = \sum_{k=1}^{K}\sum_{n=1}^{N} C\left(\gamma_k^{V2V(t)}[n]\right)$ is the sum V2V Shannon rate and $\lambda_3 = 1$. This hyperparameter selection approach balances the contribution of utility and constraint violation cost while enabling stable updates of the Q-values for improved convergence.

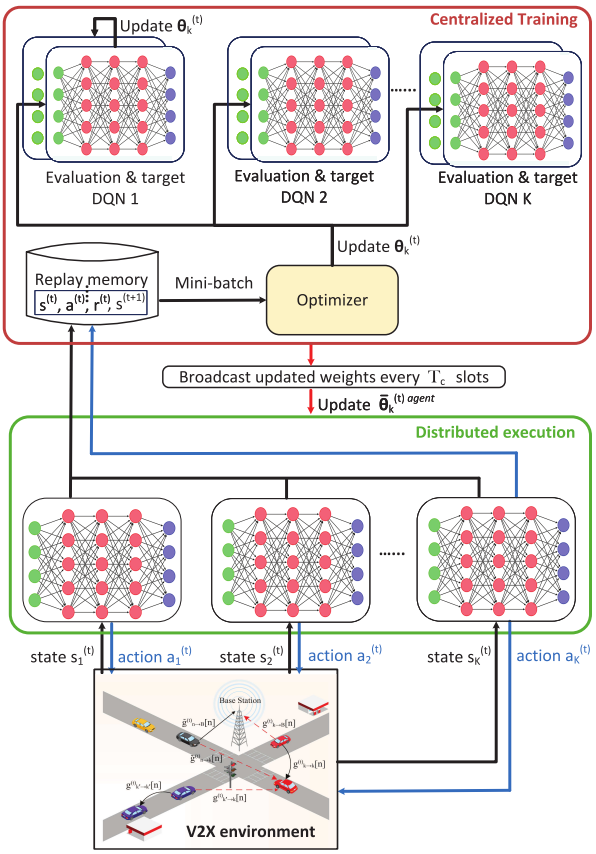


Figure 2: Illustration of the proposed multi-agent deep reinforcement learning algorithm.

B. Proposed Multi-Agent DRL-based Algorithm

We propose an MADRL scheme with each V2V transmitter as an agent. The multi-agent scenario breaks the non-stationarity assumption of the MDP as the environment may not be stationary from the agent's perspective as other learning agents update their policies. To effectively handle the non-stationarity issue in multi-agent settings, we propose the centralized training and decentralized implementation approach.

Specifically, DQNs are distributively executed at the V2V transmitters, whereas the weight parameters are centrally trained at the BS to ease implementation and improve stability.

The proposed centralized-trained distributively-executed framework is shown in Fig. (2). The centralized training and distributed execution procedures are detailed as follows:

1) ***Centralized Training****:* In the training phase, K evaluation DQNs $Q\left(s_k^{(t)}, a_k^{(t)}; \theta_k^{(t)}\right)$ with weight parameters $\theta_k^{(t)}$, and K corresponding target DQNs $Q\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta}_k^{(t)}\right)$ with weight parameters $\bar{\theta}_k^{(t)}$ are established at the BS, whereas a local DQN $Q^{agent}\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta}_k^{(t)\,agent}\right)$, with weight parameters $\bar{\theta}_k^{(t)\,agent}$ is established at the vehicle transmitter as the k-th V2V agent. The local network $Q^{agent}\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta}_k^{(t)\,agent}\right)$ has the same structure as the corresponding evaluation network $Q\left(s_k^{(t)}, a_k^{(t)}; \theta_k^{(t)}\right)$, but is established at the vehicle transmitter instead. The input and the output of each V2V agent are the local state information and the adopted local action (i.e., transmit power and spectrum allocation), respectively. In time slot t, each V2V agent k observes its current state $s_k^{(t)}$ and independently executes the action $a_k^{(t)}$ following the ϵ-greedy policy. The local observations $\left(s_k^{(t)}, a_k^{(t)}\right)$ of each V2V agent are uploaded to the BS at the beginning of time slot t, whereas the common reward $r^{(t)}$ is evaluated at the BS. Afterwards, the experience $e_k^{(t)} = \left\{s_k^{(t)}, a_k^{(t)}, r_k^{(t)}, s_k^{(t+1)}\right\}$ of each V2V agent k is stored in the replay buffer memory $\mathcal{D}$. Note that all V2V agents share the same reward in the system such that cooperative behavior among them is encouraged, and the selected experiences may include the experiences of different agents at different time slots since a single replay buffer memory is used to store experiences for all agents. In this manner, the BS trains the evaluation DQNs using the experiences gathered from all agents. Then, by sampling a mini-batch of D experiences from $\mathcal{D}$ in every T_b time slot, the trainer optimizes the weights for the DQNs by minimizing the loss function using the gradient descent technique:

$$\mathcal{L} = \frac{1}{\|D\|} \sum_{e \in D} \left(y^{(t)} - Q(s^{(t)}, a^{(t)}; \theta^{(t)})\right)^2, \tag{10}$$

where $y^{(t)} = r^{(t+1)} + \zeta \max_{a^{(t+1)}} Q\left(s^{(t+1)}, a^{(t+1)}; \bar{\theta}^{(t)}\right)$ is the target Q-value generated by the target network. In every T_c time slots, the BS updates the target DQNs with the weights of the evaluation DQNs and broadcasts the latest trained weight parameters $\theta_k^{(t)}$ of the evaluation DQN to local agent to update its weights $\bar{\theta}_k^{(t)\,agent}, \forall k \in K$. The complete training procedure is summarized in Algorithm 1, which takes the initial DQN model for each V2V agent, batch sampling period $\mathcal{T}_b$, weight copy period $\mathcal{T}_c$, and the number of train episodes $\mathcal{E}_{train}$ as the input and outputs the real-valued trained network's weight vector $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^K$ along with the background dataset $\mathcal{X}_{BG}$ to be used in Section IV for explainability purpose. Algorithm 1 achieves convergence upon completion of the training process.

2) ***Distributed Execution****:* During the distributed implementation phase, the ϵ-greedy algorithm is terminated, i.e., agents stop exploring the environment. Each V2V agent observes its environment state $s_k^{(t)}$ and executes action $a_k^{(t)}$ determined by the maximum action value according to its trained DQN, i.e., $a_k^{(t)} = \arg\max_{a_k^{(t)}} Q^{agent}\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta}_k^{(t)\,agent}\right)$. Consequently, all V2V links start transmission with the power level and spectrum sub-band determined by their selected actions. Further, the distributed execution phase outputs a hold-out dataset $\mathcal{X}$ containing the environment states and a copy of the action values (Q-values) as the labels. Note that the predicted action of a discrete action space is the index of the *argmax* function, and the experience replay already contains a copy of the action predicted by the trained network. We relegate the detailed design and use of datasets $\mathcal{X}$ and $\mathcal{X}_{BG}$ to Section IV.

Algorithm 1: Centralized Training Stage

Input: Initial DQN models, $\mathcal{T}_b$, $\mathcal{T}_c$, $\mathcal{E}_{train}$
Output: Trained weights $\boldsymbol{\theta}$, background dataset $\mathcal{X}_{BG}$

Initialize replay memory $\mathcal{D} \leftarrow \emptyset$, and $\mathcal{X}_{BG} \leftarrow \emptyset$;
Random initialize the weights $\theta_k^{(t)}$ of DQNs, and the weights of target DQNs as $\bar{\theta}_k^{(t)} = \theta_k^{(t)}, \forall k \in \mathcal{K}$;
for $e = 0 : \mathcal{E}_{train}$ **do**
  Generate the V2N and V2V communications links;
  for $t = 0 : \mathcal{T}_{steps}$ **do**
    for $k = 0 : K$ **do**
      Observe the state $s_k^{(t)}$;
      Choose the joint action $a_k^{(t)}$ following the ϵ-greedy policy based on $s_k^{(t)}$;
      Evaluate the reward $r^{(t+1)}$ in (9) and move to next state $s_k^{(t+1)}$ by executing action $a_k^{(t)}$;
      $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{s_k^{(t)}, a_k^{(t)}, r_k^{(t)}, s_k^{(t+1)}\right\}$;
      $\mathcal{X}_{BG} \leftarrow \mathcal{X}_{BG} \cup \left\{s_k^{(t)}, a_k^{(t)}\right\}$;
      if $t > T_b$ **then**
        Randomly sample a mini-batch of experiences from $\mathcal{D}$;
        Update weights $\theta_k{}^{(t)}$ by minimizing the corresponding loss function in (10) via the gradient descent technique;
        if $t \,\%\, T_c = 0$ **then**
          Copy weights $\theta_k{}^{(t)}$ of training DQN to target DQN $\bar{\theta}_k{}^{(t)}$;
          Update weights $\bar{\theta}_k{}^{(t)\,agent}$ of local agent with latest weights $\theta_k{}^{(t)}$ of training DQN;
        end
      end
    end
  end
end
return $\boldsymbol{\theta}$, $\mathcal{X}_{BG}$

IV. XAI Guided Feature Selection and Model Simplification

In this section, we introduce our XAI-based approach to simplify the trained DRL models' complexity by quantifying the relevance of input state features to the output actions for our proposed DRL-based V2X communication system. A novel post-hoc and model-agnostic framework is proposed to explain and reduce the complexity of the trained networks by eliminating some of the input variables using the state feature importance ranking.

As the basis of our methodology, we leverage the SHAP [40], a unified approach that offers a strong theoretically grounded framework for computing local and global explanations of a machine learning model. In what follows, we first introduce Shapley values, followed by a brief description

of SHAP. Then, our proposed XAI-based systematic methodology and input state feature selection algorithm for model simplification are presented.

A. Preliminaries on Shapley Values and SHAP

Feature importance rankings necessitate the computation of Shapley values which can be interpreted as the average expected marginal contribution of each input feature $x_l \in \mathbf{x}$ to a model's prediction $f(\mathbf{x})$, where $\mathbf{x}$ is the vector of original feature values. The Shapley value for x_l is computed by considering all possible coalitions of features and calculating the weighted average difference in predictions with and without the feature x_l. For a prediction model $f(\cdot)$ with input $\mathbf{x}$, the Shapley value for feature x_l is given by:

$$\phi_l(f, \mathbf{x}) = \sum_{\mathbf{z}' \subseteq \mathbf{x}'} \frac{|\mathbf{z}'|!(L - |\mathbf{z}'| - 1)!}{L!} \left[f(\mathbf{z}') - f(\mathbf{z}' \setminus \{x_l\}) \right] \quad (11)$$

where L is the total number of original features, $\mathbf{z}'$ is the subset of features used in the model, $f(\mathbf{z}')$ is the model's prediction on feature subset $\mathbf{z}'$, $f(\mathbf{z}' \setminus \{x_l\})$ is the model's prediction on feature subset $\mathbf{z}'$ excluding feature x_l. When calculating $f(\mathbf{z}')$, the x_l-th feature is masked out and then simulated by drawing random instances of the x_l-th feature from the background dataset. The Shapley value $\phi_l(f, \mathbf{x})$ signifies the impact of feature x_l on transitioning from the reference output value, i.e., the expected value without knowledge of the x_l-th feature values to the actual output provided by the prediction model. These contributions carry both magnitude and sign, allowing to assess a feature's significance. However, exactly calculating Shapley values requires searching through all possible 2^L feature combinations while retraining models, which is computationally prohibitive. As a remedy, SHAP provides a computationally efficient way to calculate Shapley values by iterating over a small subset of possible feature permutations using sampling approximations [40]. It combines previously proposed explainability methods, such as LIME [22] and Deep Learning Important FeaTures (DeepLIFT) [41], to approximate the Shapley values. As an additive feature attribution method, SHAP approximates the output $f(\mathbf{x})$ of the original complex model by summing the scores attributed to each feature, ϕ_l, as follows:

$$g(\mathbf{z}') = \phi_0 + \sum_{l=1}^{L} \phi_l z_l' \quad (12)$$

where $g(\mathbf{z}') \approx f(\mathbf{x})$ is the explanation model, and ϕ_0 corresponds to the average or expected output of the model seen during training that can be interpreted as the initial output of the model before the impact of any features is considered. Moreover, Deep Shapley additive explanation (Deep-SHAP) is an extended version designed specifically for DNNs and adapts DeepLIFT algorithm to approximate Shapley values by linearizing the non-linear components of DNNs [40]. Deep-SHAP adds explainability to the DNNs by decomposing the output prediction of DNNs on a specific input by backpropagating the contributions of all neurons to every input feature. Instead of using a single reference value, Deep-SHAP exploits a distribution of background samples such that the resulting Shapley values sum up to the difference between the expected model output on the passed background samples and the current model output.

B. Proposed Systematic XAI Methodology

The systematic XAI methodology offers two key benefits: First, it simplifies the input size by eliminating non-influential features through an iterative SHAP-based feature selection algorithm. Second, it reduces the DNN model's complexity by adjusting the architecture based on the refined input size, thereby lowering computational load and minimizing parameter updates for the distributed implementation. Our methodology can be divided into two stages. The first stage involves generating state feature importance ranking of the trained models using the Deep-SHAP explainer. The second stage uses importance-based rankings to simplify the state space of the agents using a novel variable feature selection method.

For notation consistency, we denote the k-th trained V2V agent as the prediction model, i.e., $f_k(\mathbf{x}) \equiv Q_k^{agent}(\mathbf{x})$, which takes real-valued state feature vector $\mathbf{x} = [x_1, \ldots, x_L]$ as an input, where $L = (K + 2) \times N$ is the number of input neurons. The prediction model $f_k(\mathbf{x})$ outputs the Q-value of each possible action denoted by $\mathbf{q}_k = [q_{k,1}, \ldots, q_{k,m}, \ldots, q_{k,M}]$, where $M = L_Q \times N$ is the number of output layer neurons and equal to the cardinality of the action set described in Section III-A. The trained model uses *argmax* operator to select the action maximizing the model's output.

In the first stage of generating state feature importance ranking using the trained agents, we instantiate a DeepExplainer from the SHAP library [21] and pass the trained agent's DQNs along with the background dataset to it. The background dataset $\mathcal{X}_{BG}$ is generated during the training process and serves as a prior expectation for the sample instances to be explained [42]. To compute the SHAP_values, a hold-out dataset $\mathcal{X}$ is used, which contains the state features and a copy of all possible actions, i.e., Q-values. This dataset is generated by evaluating the performance of the well-trained agents over a set of test episodes. Deep-SHAP produces local explanations in the form of importance score or "SHAP_values," $\boldsymbol{\phi}_k(\mathbf{x}) = [\phi_{k,1}, \ldots, \phi_{k,L}]$, where $\phi_{k,l}$ is the SHAP_value associated with feature $x_l \in \mathbf{x}$ and probable action $q_{k,m}$ of the k-th prediction model. To obtain global explanations, SHAP_values are obtained by repeating this across the entire database, resulting in a SHAP matrix for each sample instance in $\mathcal{X}$. The process for calculating the SHAP_values for the prediction model $f_k(\mathbf{x})$ using Deep-SHAP is delineated in Fig. 3, wherein Deep-SHAP first computes the SHAP matrix $\mathcal{S}_k$ for state sample $\mathbf{x} \in \mathcal{X}$ with one row for each action value $q_{k,m}$ and one column per state feature $x_l \in \mathbf{x}$. Then, the SHAP_values corresponding to the maximizing action index are selected from $\mathcal{S}_k$. Afterward, the absolute mean of the feature column of $\mathcal{S}_k$ is calculated across all data instances in $\mathcal{X}$. The resulting vector of mean absolute SHAP_values $\left|\boldsymbol{\phi}_k(\mathbf{x})\right| = \left[\left|\phi_{k,1}\right|, \ldots, \left|\phi_{k,L}\right|\right]$ is sorted in descending order. The first position of the resulting vector contains the most important feature, the second position contains the second most important, and so on. To reveal the meaning behind the SHAP_values in terms of their contribution to

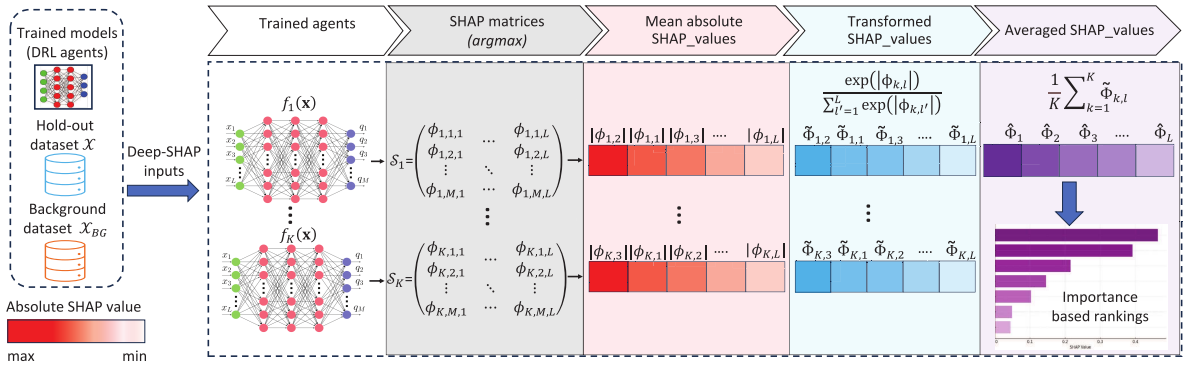


Figure 3: The process of generating SHAP_values for multiple trained models using Deep-SHAP. The pre-trained models $f_k(\mathbf{x}), \forall k \in K$ and the background dataset $\mathcal{X}_{BG}$ are utilized to calculate SHAP_values for each instance in the hold-out dataset $\mathcal{X}$. The *argmax* output helps select the appropriate vector of SHAP_values. Subsequently, the mean-absolute SHAP_values are calculated and sorted in descending order to encode feature importance. These values are then transformed using the softmax transformation. The transformed SHAP_values are averaged across the K trained models. Here, L is the total number of state features, and M is the number of valid actions.

the models' outputs, we transform the SHAP_values into a probabilistic distribution $\tilde{\mathbf{\Phi}}_k \in [0,1]^L$ through the following softmax transformation:

$$\tilde{\mathbf{\Phi}}_{k,l} = \frac{\exp\left\{\left|\phi_{k,l}\right|\right\}}{\sum_{l'=1}^{L} \exp\left\{\left|\phi_{k,l'}\right|\right\}}, \forall k \in \mathcal{K}, l = 1, \ldots, L, \qquad (13)$$

where $\tilde{\mathbf{\Phi}}_{k,l} = \left[\tilde{\Phi}_{k,1}, \ldots, \tilde{\Phi}_{k,L}\right]$ denotes the real-valued vector of transformed SHAP_values for the k-th prediction model.

To obtain global state feature importance ranking, the transformed absolute SHAP_values are obtained for each trained agent using the above procedure and averaged across the K trained models. The averaged transformed SHAP_values, denoted by $\widehat{\mathbf{\Phi}} = \left[\widehat{\Phi}_1, \ldots, \widehat{\Phi}_L\right]$ are utilized to rank the input state features, which can be interpreted as global state-feature importance scores. Then, by utilizing an appropriate feature selection strategy, the most important state elements contributing to the model outcome are selected, and the model is retrained using the subset of these important states. Note that the presented systematic feature selection strategy in Fig. 3 is model-agnostic and applicable to any other type of pre-trained DRL-based radio resource management system.

Algorithm 2: Iterative SHAP-based Feature Selection

Input: $f_{k\,(\forall k \in K)}$, $\mathcal{X}$, $\mathcal{X}_{BG}$, Δ
Output: Subset of input state features
1 **for** $k = 0, 1, \ldots, K$ **do**
2 Generate SHAP matrix $\mathcal{S}_k$ for $f_k(\mathbf{x})$ using $\mathcal{X}$ and $\mathcal{X}_{BG}$;
3 Obtain transformed SHAP_values $\tilde{\mathbf{\Phi}}_{k,l}$ using (13);
4 **end**
5 $\widehat{\mathbf{\Phi}} \leftarrow$ Average $\tilde{\mathbf{\Phi}}_{k,l}$ across the K agents;
6 Generate global state-feature importance rankings using $\widehat{\mathbf{\Phi}}$;
7 $\alpha_{original} \leftarrow$ Evaluate the average network performance on $\mathcal{X}$ using (14);
8 $p \leftarrow 1$;
9 **while** $p < L$ **do**
10 Mask the values of the p least important features within $\mathcal{X}$;
11 $\alpha_{simplified} \leftarrow$Re-evaluate the average network performance using the partially masked dataset ;
12 **if** $|\alpha_{original} - \alpha_{simplified}| \geq \Delta$ **then**
13 Eliminate the $p-1$ least important features;
14 **return** *Subset of input state features*;
15 **end**
16 **else**
17 $p \leftarrow p + 1$;
18 **end**
19 **end**
20 Eliminate the $p-1$ least important features;
21 **return** *Subset of input state features*

C. Post-hoc SHAP-based Input State Selection Algorithm

We devise an iterative state feature selection algorithm based on SHAP-based state feature importance rankings. The post-hoc SHAP-based state feature selection strategy is summarized in Algorithm 2, whose inputs are the K trained agents, hold-out dataset $\mathcal{X}$, background dataset $\mathcal{X}_{\text{BG}}$, and the precision threshold Δ, whereas the outputs are the optimal subset of features retained using the proposed feature selection strategy. The developed variable feature selection strategy selects the number of important state features by repeatedly evaluating the trained models based on a predefined metric while masking step-by-step the values of the p least important features according to the given importance-based feature ranking. Significant changes in the evaluation results indicate that a sufficiently important state feature has been randomized in the current step. Consequently, all state features randomized in the previous step can be confidently removed from the trained network's input.

While the network is trained to maximize cumulative reward in (9), this reward is not the most illustrative way to evaluate its post-hoc performance due to its high variance. To evaluate the trained models' performance, we consider the following reward value averaged across the hold-out dataset

$$\alpha = \frac{1}{\|\mathcal{X}\|} \sum_{i=1}^{\|\mathcal{X}\|} r^{(i)}, \qquad (14)$$

where $r^{(i)}$ is calculated using the i-th state-action tuple in the hold-out data set, $\mathcal{X}$, as per (9).

Algorithm 2 starts with a for loop (Lines 1 – 4) that iteratively computes the SHAP matrix $\mathcal{S}_k, \forall k \in K$, and obtains

transformed SHAP_values, $\tilde{\mathbf{\Phi}}_{k,l}$, $\forall k \in K$, as per (13). Then, $\tilde{\mathbf{\Phi}}_{k,l}$, $\forall k \in K$, are averaged across the trained models and utilized to rank the input state features (Lines 5 − 6). The generated feature rankings are used to select a subset of state features using the average network performance metric in (14). The algorithm first evaluates average network performance in (14) with no state features removed, denoted by $\alpha_{original}$ (Line 7). Then, the p least important features are masked according to SHAP importance-based feature ranking, i.e., their values are changed to the respective column variance (Line 10). The trained models are re-evaluated on the modified dataset, and the evaluation results are denoted as $\alpha_{simplified}$ (Line 11). If there is a significant change in the evaluation results, i.e., $\left|\alpha_{original} - \alpha_{simplified}\right| \geq \Delta$, where Δ is the precision threshold, then the $p-1$ least important state features are removed and only $L - p + 1$ influential state features are retained (Lines 12−15). Otherwise, the value of p is incremented, and the state feature masking and re-evaluation steps previously defined are performed iteratively (Lines 17−19). It should be noted that the value of Δ is set to be between 1-10% of the average network performance metric, ensuring that the algorithm outputs only a subset of importance-based features.

One major obstacle associated with the Deep-SHAP based SHAP_values estimation is the considerable computational complexity, which increases exponentially with the number of state features and linearly with the number of background data samples [40]. Our initial experimentation shows that utilizing the whole training dataset $\mathcal{X}_{\text{BG}}$ significantly slows down the SHAP computations. To strike a balance between achieving high accuracy in SHAP_values estimates and keeping computation time reasonable, we utilize the transformation in the existing SHAP implementation by K-means to summarize $\mathcal{X}_{BG}$ [21]. Specifically, we use 1% K-means summary, i.e., 4,000 samples of $\mathcal{X}_{BG}$ and 10% K-means summary, i.e., 1,000 test samples, to serve as the background and hold-out dataset, respectively. This summarization is sufficient to accurately assign state feature importance to different input states without incurring the full computational burden of calculating SHAP_values over all data samples in the background dataset.

Note that the fast-changing nature of the state features does not pose a problem for the proposed XAI method. The post-training inference process can be performed in an environment twin (a simulation of the real environment), and the feature selection process is fast enough so that the environment does not change in the meantime. Additionally, our XAI-based feature selection process does not interact with the environment as it only utilizes the already-trained DRL agents and stored environment observations.

V. Performance Evaluation

This section evaluates the performance of the proposed MADRL resource allocation scheme in a single-cell V2X communications system. Numerical results are presented to validate the effectiveness of applying the XAI-based model simplification strategy.

The following algorithms are simulated for performance comparison to verify the efficiency of our proposed XAI-based simplified multi-agent DRL (Simplified-MADRL) scheme.

1) **Original multi-agent DRL (Original-MADRL)**: This algorithm trains the multiple V2V agents using the proposed centralized training with a decentralized execution framework described in Section III-B. This algorithm considers the original full state features of the multiple agents during the training and execution process, hence named Original-MADRL.
2) **Centralized Single-agent DRL (SADRL)**: This algorithm trains a single meta-agent at the BS, which outputs joint actions for all V2V agents. Since a single DQN jointly handles spectrum sub-band selection and transmit power control, the complexity of the output layer size and state action pairs to be visited for convergence is proportional to the total number of V2V pairs in the network. Apart from the complexity, this algorithm is executed in a centralized way with accurate global CSI availability requirements.
3) **Full power**: This algorithm selects the channel with the lowest interference for V2V transmission, setting the transmit power to the maximum level $P_{\max}$ for all V2V links. The computational complexity is dominated by the enumeration of all N available spectrum sub-bands, where a fast-sorting algorithm is used centrally at the BS to efficiently compare interference levels across the sub-bands and identify the one with the least interference [14].
4) **Random allocation**: This algorithm chooses a random transmit power and spectrum sub-band action at the beginning of each time step.

For the learning algorithms, we consider an episodic setting with each episode lasting for $\mathcal{T}_{steps}$ =100 time steps. At the beginning of each episode, the environment state is initialized randomly, determined by the initial transmission powers of vehicular links and environment states. The training phase lasts $\mathcal{E}_{train}$ = 4000 episodes, whereas $\mathcal{E}_{test}$ = 1000 episodes are dedicated to the testing phase, where the well-trained DNNs are utilized to evaluate the performance of the different learning algorithms.

All simulations are performed on a 10-Core Intel(R) Xenon(R) Silver 4114, 2.2GHz system equipped with an Nvidia Quadro P2000 graphics processing unit (GPU). The V2X simulation environment and algorithms are developed in Python, and the DNNs are built and trained using Tensorflow API. The open-source implementation of SHAP [21] is used to compute explanations for the models.

A. Simulation Setup

We follow the simulation setup for V2X communications in annex A 3GPP TR 36.885, which describes in detail the vehicular channel models, vehicle mobility, and vehicular data traffic for an urban scenario [34]. The urban scenario considers an area of size 750x1299 meters with 9 buildings and roads between buildings, with 2 lanes in each direction for each road. Vehicles are moving in this area at equal constant speed. N V2N links are initiated by N vehicles and the K V2V links are formed between each vehicle with its surrounding neighbors. The V2N links follow the path-loss model $PL(d) = 128.1 + 37.6\log_{10}(d)$, where d is the

Table II: Simulation parameters

Parameter	Value
Carrier frequency	2 GHz
Maximum transmit power $P_{\max}$	23 dBm
Noise power σ^2	-114 dBm
Vehicle receiver noise figure	9 dB
Vehicle placement model	spatial Poisson
Carrier frequency f_c	2 GHz
Shadowing standard deviation σ_{sh}	3 dB
BS antenna height	25 m
BS antenna gain	8 dBi
BS receiver noise figure	5 dB
Distance between RSU and highway	35 m
Vehicle antenna height	1.5 m
Vehicle antenna gain	3 dBi
Payload size L	100 bits
Batch sampling period $\mathcal{T}_b$	100
Weight copy period $\mathcal{T}_c$	400

Table III: Hyperparameters for Deep-Q Networks

Layers	Input	layer 1	layer 2	layer 3	Output
Neuron number	$(2+K)\times N$	$5\|\|\mathbf{x}\|\|+8$	$3\|\|\mathbf{x}\|\|$	$2\|\|\mathbf{x}\|\|$	$L_Q \times N$
Activation function	Linear	Tanh	ReLU	Tanh	ReLU
Mini-batch size D	100				
Optimizer	RMSProp				
Discount factor ζ	0.99				
Initial learning rate $\delta^{(0)}$	0.01				
Learning decay rate β_l	0.0001				
Initial exploration rate $\epsilon^{(0)}$	0.1				
ϵ-decay rate β_ϵ	0.0001				

distance in [km] of the vehicle from the BS [43]. For the V2V links, we use the WINNER path-loss model $PL(d) = A\log_{10}(d) + B + C\log_{10}\left(\frac{f_c}{5.0}\right)$, where d is the inter-vehicular distance in [m], f_c is the system frequency in [GHz], the parameter A describes the impact of path-loss exponent, the parameter B is the intercept, and parameter C controls the frequency dependency. We set $\omega_1 = \omega_2 = 1$ in (7) to consider a fair and equal contribution of V2N and V2V link's sum rates. The parameter values for A, B, C are calculated according to the guidelines (Table 4-4) provided for $B1$ urban micro-cell scenario [43]. To achieve the demand of the URLLC, we set the packet transmission latency Δ_T to 1 ms, and the bandwidth occupied by each V2N link B_w to 180 KHz. Other important simulation parameters are provided in Table II.

The hyperparameter values for the DQNs are tabulated in Table III. The tabulated parameters are fine-tuned to reach the desired results and chosen as the final settings because they consistently deliver the best performance across multiple experimental trials. The initial learning rate $\delta^{(0)} = 0.01$ is annealed as $\delta^{(t+1)} = (1-\beta_l)\delta^{(t)}$, with $\beta_l \in (0,1)$ as the decay rate. The discount factor $\zeta = 0.99$ ensures higher cumulative rewards, benefiting V2V/V2N data transmission. The ϵ-greedy algorithm starts with $\epsilon^{(0)} = 0.1$ and adapts as $\epsilon^{(t+1)} = \max 0, (1-\beta_\epsilon)\epsilon^{(t)}$, where $\beta_\epsilon = 0.0001$. Batch size $D = 100$ balances update noise and computational efficiency. The DNNs used to implement the DQN consist of one input layer, three fully connected hidden layers, and one output layer. The input layer processes a state vector of length $|\mathbf{x}| = (2+K)\times N$, while the hidden layers have $H_1 = 5|\mathbf{x}| + 8$, $H_2 = 3|\mathbf{x}|$, and $H_3 = 2|\mathbf{x}|$ neurons. This architectural design ingenuity reduces model size based on input feature reduction, balancing computational efficiency and accuracy. The rectifier linear unit (ReLU) and hyperbolic tangent (Tanh) functions are used alternatively as the activation function since this scheme leads to faster convergence, as per our observation.

B. Analysis of XAI-based Variable State Feature Selection

The main objective of our work is to simplify the proposed DRL-based framework through importance-based feature selection without losing performance. We first investigate the efficiency of the proposed importance-based variable feature selection methodology. Notice that the importance-based feature selection is a post-hoc process whereby the agents are first trained and tested using the Original-MADRL framework. Then, the summarized datasets $\mathcal{X}_{BG}$, $\mathcal{X}$ obtained from the training and execution process are utilized to obtain global state feature importance rankings using the systematic XAI-based methodology described in Section IV-B.

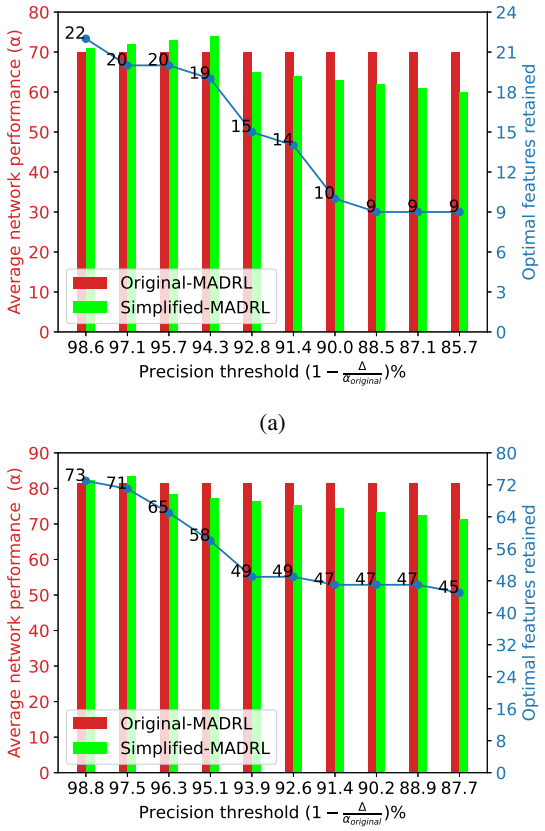


Figure 4: Average network performance and retained state features versus the precision threshold, with: (a) $K = N = 4$ where the Original-MADRL network has 24 state features per agent, and (b) $K = N = 8$ where the Original-MADRL network has 80 state features per agent.

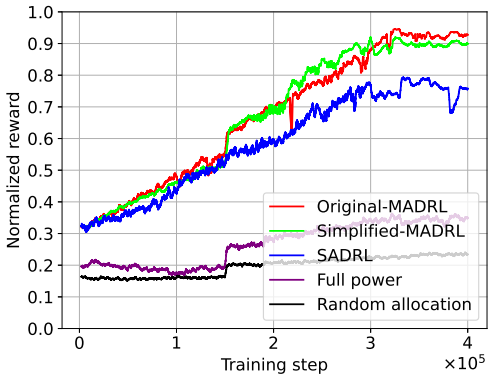
Fig. 4a shows the average network performance α and importance-based retained state features for different Δ values. Considering different QoS requirements for the V2V agents, we set different error probability constraints for each vehicle, $\varepsilon_{\max,k} = [10^{-4}, 10^{-3}, 10^{-4}, 10^{-3}]$. It can be observed that an increase in Δ allows for retaining fewer optimal features. As

seen, for $\Delta \leq 2$, the Simplified-MADRL network does not lose any performance compared to the Original-MADRL network. In fact, a slight performance improvement is observed due to the elimination of less important features that are most likely non-informative and add complexity to the network without providing performance benefits. For $\Delta > 2$, there is a sharp decrease in the number of state elements, where approximately 38% of the original state features are eliminated. This sharp decline is reflected by the decrease in α, showing that some of the influential state input features have been removed.

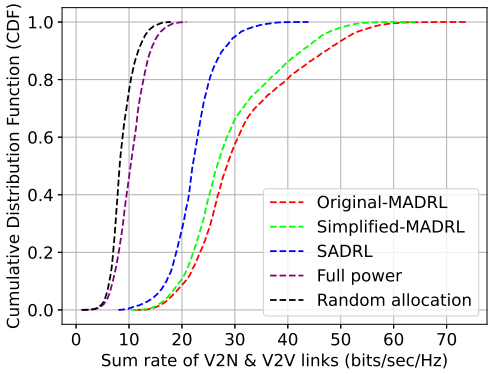
Fig. 4b shows the average network performance and retained state features for different Δ values in a network with $N = 8$ V2N links and $K = 8$ V2V pairs. Here, we set error probability constraints as $\varepsilon_{\max,k} = [10^{-3}, 10^{-4}, 10^{-2}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-3}, 10^{-4}]$. Similar to the case for four V2V links, it can be observed that an increase in Δ results in fewer optimal state feature selections. For $\Delta = 2$, the proposed variable feature selection methodology selects 58 optimal state features out of 80 for the Original-MADRL network while maintaining the average network performance gap less than ~5% between the Original-MADRL scheme and the Simplified-MADRL scheme, respectively. Based on the results from Fig. 4a, and 4b, there exists a trade-off between model input simplification (e.g., via feature selection) and network performance. To strike the right balance between feature selection and optimal performance in terms of the average network performance metric, a moderate value of $\Delta = 2$ is selected to achieve performance close to the Original-MADRL network while reducing the number of optimal state features by ~21% and ~28% for a network with K=4 and K=8 V2V pairs, respectively. Therefore, for the subsequent simulation results, we fix the value of the precision threshold to $\Delta = 2$.

C. Performance Comparison of Algorithms

1) Convergence Analysis and Sum rate performance: Fig. 5a illustrates the training convergence behavior of the different algorithms in terms of the reward performance in a network with 4 V2V links and V2N links. It can be observed that the Original-MADRL scheme and the Simplified-MARL schemes converge around 3.2×10^5 and 2.8×10^5 time steps, respectively. In contrast, the centralized SADRL scheme demonstrates slow and suboptimal convergence, primarily due to the high complexity of its action space, where a single DQN determines power and sub-band allocation actions for all V2V links, rendering the approach impractical due to the significant transmission overhead involved. The Full power algorithm exhibits degraded performance as it always transmits at maximum power. Such resource allocation can lead to a lower sum rate and violation of reliability constraint if multiple links use the same sub-band. Moreover, the Full power algorithm does not consider adapting the transmit power based on the link quality. The random allocation exhibits the worst performance because it randomly selects the sub-band and transmits power at each time step, irrespective of the link quality or available resources. The proposed XAI-based Simplified-MADRL scheme utilizing only 19 state features



(a) Training.

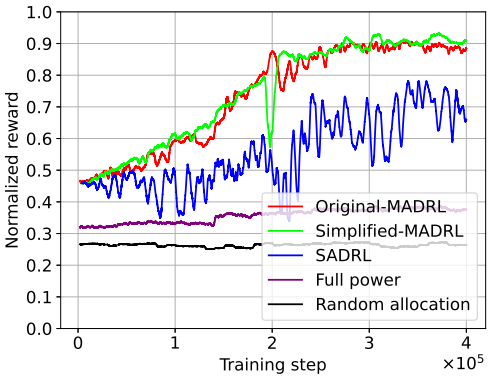


(b) Testing- Empirical CDF.

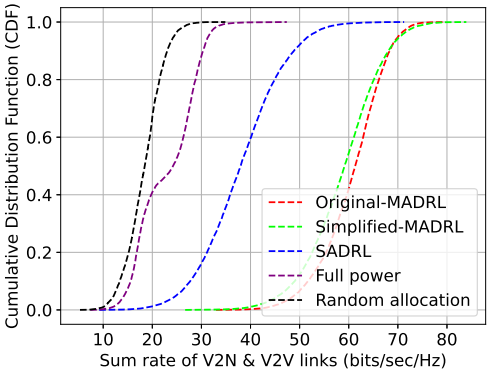
Figure 5: Training and testing performance of network with $K = N = 4$, $\boldsymbol{\varepsilon}_{\max} = [\varepsilon_{\max,1}, \ldots, \varepsilon_{\max,k}, \ldots, \varepsilon_{\max,K}] = [10^{-4}, 10^{-3}, 10^{-4}, 10^{-3}]$, where (a) shows the learning process comparison for the different algorithms, (b) shows the empirical CDF of network sum-rate performance. Each value is the moving average of the previous 300 time slots.

can remarkably converge to ~97% of the training performance of the Original-MADRL using 24 state feature inputs. This means that the proposed variable selection strategy can effectively filter the most relevant features contributing to the agents' actions without significant performance loss.

Fig. 5b shows the cumulative distribution function (CDF) of the achievable sum rate for the V2V and V2N links, comparing the performance of different algorithms during the deployment (inference) phase in a network with $K = N = 4$ V2V and V2N links. Fig. 5b further demonstrates the effectiveness of the proposed Simplified-MARL algorithm by empirically validating its performance against the different benchmark algorithms in the inference stage. The Simplified-MARL scheme outperforms the SADRL scheme and achieves on average 41% higher achievable sum-rates for all the V2V and V2N links compared to the SADRL scheme. Further, the average difference between the Simplified-MARL and the Original-MADRL schemes in terms of the sum-rate performance is less than ~7%. This means that by exploiting the well-trained DNNs, the proposed algorithm can effectively learn



(a) Training.



(b) Testing- Empirical CDF.

Figure 6: Training and testing performance of network with $K = N = 8$, $\boldsymbol{\varepsilon}_{\max} = [\varepsilon_{\max,1}, \ldots, \varepsilon_{\max,k}, \ldots, \varepsilon_{\max,K}] = [10^{-3}, 10^{-4}, 10^{-2}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-3}, 10^{-4}]$, where (a) shows the learning process comparison for the different algorithms, (b) shows the Empirical CDF of network sum-rate performance. Each value is the moving average of the previous 300 time slots.

the patterns of the environment based on the optimal state features without requiring the full set of state feature inputs.

Fig. 6a illustrates the training convergence behavior of different algorithms in a network with $K = N = 8$ V2V and V2N links. It is observed that the rewards are low and unstable at the beginning of the learning as the training agents focus more on exploration than exploitation. Despite fluctuations due to mobility-induced channel fading in vehicular environments, the reward function increases and eventually converges with the increasing number of training iterations. The Simplified-MADRL and the Original-MADRL schemes have comparable training performance, and both schemes converge in about 2.8×10^5 training steps. The SADRL scheme fails to achieve comparable performance to the proposed method and demonstrates reduced training efficiency under higher vehicular densities. This inadequacy is attributed to the rapid growth of the DQN network topology and action space as the vehicular network scales, making it challenging to thoroughly explore the entire action space to determine the optimal action selection.

Similar to Fig. 5b, Fig. 6b shows the CDF of the achievable

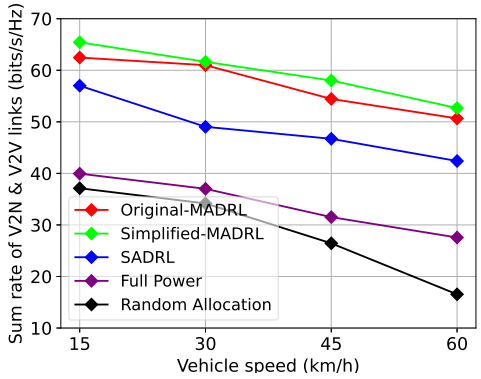


Figure 7: Sum rate performance of V2N and normal V2V links with varying vehicle speeds.

sum-rate for the V2V and V2N links; however, in a network with $K = N = 8$ V2V and V2N links. The proposed approach outperforms the SADRL scheme, achieving, on average, a 66% higher sum-rate in the testing stage. Interestingly, the Simplified-MADRL achieves higher sum-rates for the majority of the V2N and V2V links than the Original-MADRL scheme with ~28% fewer state features. This improvement is mainly attributed to eliminating less important and non-informative features that only complicate the learning process.

Note that compared to the recursive feature selection strategies requiring retraining for different subsets of features, the desired results for the Simplified-MARL algorithm are achieved after a single retraining of the models on the subset of important features predicted by Algorithm 2. As seen from Fig 5b, 6b, the simplified agents have not lost significant performance when compared to the original agents' performance. In fact, a slight performance improvement in learning efficiency for larger network size is observed in Fig 6 due to eliminating less important features. This demonstrates that the proposed feature selection strategy effectively identifies the most relevant features for the multiple agent's actions with minimal impact on performance.

2) Impact of Mobility: Fig. 7 shows the impact of vehicle speed on the sum data rate of V2N and V2V links, when the number of V2V and V2N links are set to $K = N = 8$. From the figure, a general decline in sum rate performance is observed across all approaches as vehicle speed increases. This drop in performance is primarily due to reduced received power in V2X communication links, resulting from increased inter-vehicle distances at higher speeds, which lead to sparser traffic conditions. Additionally, higher speeds create observation uncertainties (e.g., channel state information and received interference), making it difficult to find an optimal policy, especially the random search approach, which shows the lowest performance. Our proposed Simplified-MADRL approach performs close to the Original-MADRL solution considering all state features during the learning process. Furthermore, the proposed MADRL-based solutions outperform the centralized SADRL and the Full power algorithms in terms of sum-rate performance across various vehicle speeds, adapting more effectively to dynamic networks.

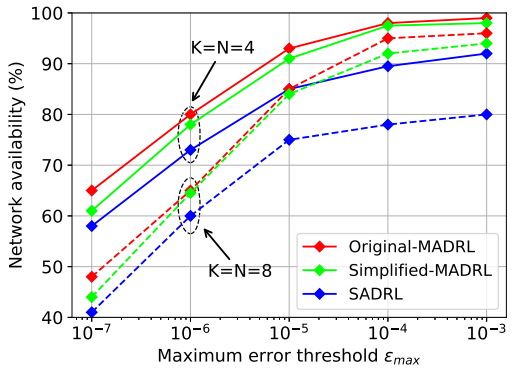


Figure 8: Network availability for different maximum error probability thresholds.

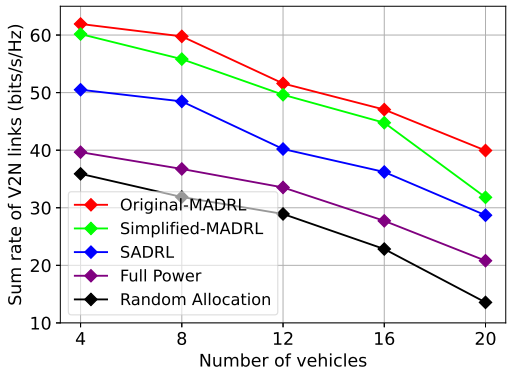


Figure 9: Sum rate of V2N links with varying number of vehicles.

3) Reliability performance: In Fig. 8, we consider the impact of reliability constraint violation. Thus, the performance metric is the network availability [44], defined as the percentage of the total number of channel generations in the testing phase, for which the worst-case decoding error probability, i.e., $\max_{k\in K} \varepsilon_k^{(t)}[n]$ is no greater than a threshold $\varepsilon_{\max}$. Here, we keep the same maximum threshold error probability for all V2V links, i.e, $\varepsilon_{\max,k} = \varepsilon_{\max}\ \forall k \in K$. Across all schemes, the network availability percentage improves with increasing $\varepsilon_{\max}$ as the tolerable decoding error probability requirement becomes less stringent. Original-MADRL consistently achieves the highest reliability, reaching ~92% at $\varepsilon_{\max} = 10^{-5}$ for $K = N = 4$ and nearly 98.3% at $\varepsilon_{\max} = 10^{-4}$. The proposed Simplified-MADRL follows closely, with only a slight gap below Original-MADRL, achieving around 98% at $\varepsilon_{\max} = 10^{-4}$ for $K = N = 4$. Network availability decreases as the number of V2V and V2N links increases due to fixed resource sharing and higher interference, worsening reliability violations. Simplified-MADRL maintains reasonable reliability for $\varepsilon_{\max} > 10^{-5}$, performing close to Original-MADRL. In contrast, centralized SADRL falls behind, achieving only $\sim$ 80% for $K = N = 8$ at $\varepsilon_{\max} = 10^{-3}$. The gap is largest at lower error levels, highlighting the resilience of Original-MADRL and Simplified-MADRL to increased V2V link density.

4) Scalability: Fig. 9 demonstrates the impact of varying number of vehicles on the sum rate of the V2N links. Here, we fix the number of sub-bands for V2N links to $N = 4$ and increase the number of V2V links to assess the impact of dynamic vehicle numbers on the sum-rate performance. From the figure, it is observed that when the number of vehicles increases, more V2V links share the fixed number of sub-bands, which causes stronger interference to the V2N links, resulting in a rate reduction for all schemes. The proposed Simplified-MADRL approach achieves better performance than the other baseline schemes, demonstrating its robustness across different numbers of vehicles. The centralized SADRL approach shows degraded performance mainly caused by the increased size of the joint learning action space and increased DQN output layer complexity, whereas the poor performance of the Full power algorithm can be explained by the strong interference due to maximum power transmission, negatively impacting the sum rate of the V2N links.

D. Computational Complexity Reduction Analysis

The computational complexity of the proposed learning algorithms depends on the structure of implemented DNNs (number of hidden layers and the number of corresponding neurons). The complexity of learning algorithms is primarily dominated by the update process of the network weight parameters. Each V2V agent requires a periodic update of its local DQN parameters during training. The input layer of each DQN always matches the number of input features, and the number of neurons in the hidden layers depends on these input features. Therefore, reducing the number of input features decreases the computational load for propagating information through the network and also reduces the number of parameters that need to be periodically broadcasted to update the local agents' DQNs.

The total number of scalar parameters to broadcast per agent is $|\theta| = \left(\|\mathbf{x}\| + \sum_{l=1}^{3} H_l H_{l+1}\right)$. The update process of the loss function is linear to the mini-batch size D since the optimizer scans all mini-batch samples. As a result, an update of all parameters incurs a computational complexity of $O\left(D(\|\mathbf{x}\| + \sum_{l=1}^{3} H_l H_{l+1})\right)$. Considering the proposed Simplified-MADRL approach, the overall computational complexity is $O\left(KD(\|\mathbf{x}\| + \sum_{l=1}^{3} H_l H_{l+1})\right)$ for updating the network parameters of all K agents. Table (IV) shows the complexity comparison in terms of the average training time, the number of parameters to broadcast, and the number of state features used according to the variable feature selection methodology. Interestingly, the simplified-MADRL reduces average training time computed across training steps by ~26% and ~11% for K=4 and K=8 V2V agents, respectively. In terms of the the broadcast parameters per agent, the proposed approach achieves a reduction of ~35% and ~46% by utilizing 19 and 58 most important state features, compared to the 24 and 80 state features used by the Original-MARL agent. In our evaluation environment, the agents spend an average of 8.3×10^{-4} seconds for each action selection. There is considerable potential

Table IV: Complexity comparison of algorithms.

Scheme	Average Training Time (ms)		Number of broadcast parameters		Number of state features	
	K=4	K=8	K=4	K=8	K=4	K=8
Original-MADRL	6.23	13.4	34,544	347,920	24	80
Simplified-MADRL	4.56	11.9	22,424	185,912	19	58

for optimizing execution efficiency since the computational capabilities of processing units in real-world V2X networks exceed those of the computers utilized in lations.

E. Discussion on Implementation and Future Extensions:

Practical implications: The proposed MADRL-based solution for spectrum and power resource allocation requires an offline training phase and a deployment phase. Offline training is employed to mitigate the computational overhead of learning the mapping between input states and output actions. Subsequently, during online network deployment, this mapping is executed using the pre-trained DNNs, bypassing the need for intensive computations. Offline training can be performed using a digital twin of the V2X network, incorporating network topology, channel models, and QoS requirements. The proposed algorithm can utilize this twin for offline initialization and network training at the BS to explore the simulated environment, mitigating real-world risk. Further, in dynamic vehicular networks, it is challenging to provide theoretical convergence guarantees for the proposed MADRL-based solution and other advanced multi-agent adaptations, despite their good empirical performances [45]. By giving all agents a common global reward, the proposed MADRL-based solution mitigates the instability of the multi-agent environment but makes each agent fail to achieve the higher individual reward. The proposed XAI-based framework could be easily adapted to work with other multi-agent frameworks and cooperative value decomposition-based methods such as QMIX [46] and its adaptations to improve convergence stability and cooperation among the agents.

Dynamic network topology: It is worth pointing out that the well-trained DRL models can be easily outdated due to the high mobility of vehicles. In our proposed centralized training and decentralized implementation approach, the weight parameters of the DNNs are centrally trained at the BS. The BS periodically broadcasts the latest trained weight parameters to the local agents. Retraining can be triggered if the network performance falls below a threshold. Further, considering nearby V2V pairs often experience similar channel quality and environment observations, the RSU can intelligently cluster vehicles into a specific zone and use the zone information to train newly activated V2V pairs. For newly activated V2V pairs, they request the BS for connectivity. The RSU has complete information on the vehicles' location and network topology. For an incoming vehicle, the RSU establishes the corresponding DNNs and transfers the weights of the already-trained model to the newly established DNNs, which are downloaded upon request from the incoming vehicle. To accommodate the changing number of V2V agents, state encoding adjusts for changes in the number of V2V agents under RSU coverage. If the number of vehicles exceeds the maximum number of trained agents, information from the extra vehicles is not included in the state. Moreover, if one or more vehicles leave the coverage area, the RSU can introduce virtual agents with their direct and interfering channel gains set to zero to avoid impacting other agents' decision-making. The purpose of having these virtual agents as placeholders is to provide inconsequential inputs to fill the input elements of fixed length, i.e., the agent's state is padded with zeros. Further, the proposed XAI-based feature selection strategy somehow mitigates this issue since all trained agents use a fixed size of importance-based state features during training and testing.

Generalization to different environments: The proposed method is designed to adapt to the specific characteristics of a given vehicular environment. However, when the environment changes, the DNNs need to be retrained, and the feature selection needs to be reevaluated. In such cases, the multiple agents can be trained from scratch or initialized with the existing DNN weights, then refined using data from the new environment. To enhance the adaptability of the proposed MADRL-based solution to new environment scenarios and tasks requiring computation-intensive training, model-agnostic meta-learning [16], [47] can be incorporated into the proposed framework.

VI. Conclusions

In this paper, we have developed a model-agnostic XAI-based methodology to explain the inference process of multiple trained DRL agents and have used it to simplify the DRL agents' input state size through importance-based feature selection. We have capitalized on SHAP for generating feature importance explanations and reducing the complexity of multiple trained models by generating a new model with a reduced size that requires lower training time and reduced network parameters to broadcast while maintaining a reasonable performance. We have applied the proposed methodology to a practical vehicular communication network where multiple agents are first trained using a novel centralized learning and decentralized execution algorithm. Subsequently, the XAI-based framework is applied to the trained models to select important features using an automated feature selection strategy. Our results have shown that the XAI-assisted methodology remarkably reduces the optimal state features by ~21% and ~28%, average training time by ~26% and ~11%, and trainable weight parameters to broadcast by ~35% and ~46% in a network setting with four and eight vehicular pairs, respectively. In the future, we plan to adapt the proposed XAI-based methodology for other wireless resource allocation problems. Additionally, applying XAI to digital twin systems to simplify complex virtual representations of real-world networks and maintain synchronization between the twin and the actual network is a promising research direction. By interpreting model actions using environment semantics, XAI can enhance the connection between digital twins and real-world dynamics.

References

[1] H. Bagheri, M. Noor-A-Rahim *et al.*, "5G NR-V2X: Toward connected and cooperative autonomous driving," *IEEE Commun. Standards Mag.*, vol. 5, no. 1, pp. 48–54, 2021.

[2] M. H. C. Garcia *et al.*, "A tutorial on 5G NR V2X communications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1972–2026, 2021.

[3] A. Celik and A. M. Eltawil, "At the dawn of generative AI era: A tutorial-cum-survey on new frontiers in 6G wireless intelligence," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2433–2489, 2024.

[4] M. Parvini *et al.*, "Resource allocation in V2X networks: From classical optimization to machine learning-based solutions," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 1958–1974, 2024.

[5] N. Khan, S. Coleri, A. Abdallah, A. Celik, and A. M. Eltawil, "Explainable and robust artificial intelligence for trustworthy resource management in 6G networks," *IEEE Commun. Mag.*, vol. 62, no. 4, pp. 50–56, 2024.

[6] L. Liang *et al.*, "Graph-based resource sharing in vehicular communication," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4579–4592, 2018.

[7] T. Xue *et al.*, "Two-timescale vehicle association and resource management for C-V2X networks," *IEEE Commun. Lett.*, vol. 12, no. 7, pp. 1259–1263, 2023.

[8] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for D2D-enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, 2017.

[9] P. Wang *et al.*, "Joint spectrum and power allocation for V2X communications with imperfect CSI," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16 338–16 353, 2023.

[10] P. Xiang *et al.*, "Multi-agent RL enables decentralized spectrum access in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10 750–10 762, 2021.

[11] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, 2019.

[12] J. Tian *et al.*, "Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous QoS guarantees for vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1683–1695, 2022.

[13] A. S. Kumar, L. Zhao, and X. Fernando, "Multi-agent deep reinforcement learning-empowered channel allocation in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1726–1736, 2021.

[14] X. Li *et al.*, "Federated multi-agent deep reinforcement learning for resource allocation of Vehicle-to-Vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8810–8824, 2022.

[15] X. Zhang *et al.*, "Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6380–6391, 2020.

[16] Y. Yuan *et al.*, "Meta-reinforcement learning based resource allocation for dynamic V2X communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8964–8977, 2021.

[17] P. Liu, H. Cui, and N. Zhang, "DDQN-based centralized spectrum allocation and distributed power control for V2X communications," *IEEE Trans. Veh. Technol.*, pp. 1–12, 2024.

[18] B. Hazarika *et al.*, "Radit: Resource allocation in digital twin-driven UAV-aided internet of vehicle networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3369–3385, 2023.

[19] C. Li, W. Guo *et al.*, "Trustworthy deep learning in 6G-enabled mass autonomy: From concept to quality-of-trust key performance indicators," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 112–121, 2020.

[20] W. Guo, "Explainable artificial intelligence for 6G: Improving trust between human and machine," *IEEE Commun. Mag.*, vol. 58, pp. 39–45, 2020.

[21] S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," 2016. [Online]. Available: https://arxiv.org/abs/1602.04938

[23] A. Terra, R. Inam *et al.*, "Explainability methods for identifying root-cause of SLA violation prediction in 5G network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. IEEE, 2020, pp. 1–7.

[24] P. Barnard *et al.*, "Resource reservation in sliced networks: An explainable artificial intelligence (XAI) approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2022, pp. 1530–1535.

[25] P. Barnard, N. Marchetti, and L. A. DaSilva, "Robust network intrusion detection through explainable artificial intelligence (XAI)," *IEEE Netw. Lett.*, vol. 4, no. 3, pp. 167–171, 2022.

[26] S. Ben Saad, B. Brik, and A. Ksentini, "A trust and explainable federated deep learning framework in zero touch B5G networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 1037–1042.

[27] F. Rezazadeh, H. Chergui, and J. Mangues-Bafalluy, "Explanation-guided deep reinforcement learning for trustworthy 6G RAN slicing," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2023, pp. 1026–1031.

[28] G. Rjoub, J. Bentahar, and O. A. Wahab, "Explainable AI-based federated deep reinforcement learning for trusted autonomous driving," in *Proc. Int. Wireless Commun. and Mobile Comput. (IWCMC)*, 2022, pp. 318–323.

[29] S. Bhandari *et al.*, "Feature selection improves tree-based classification for wireless intrusion detection," in *Proc. 3rd Int. Workshop Syst. Network Telemetry Anal.*, 2020, pp. 19–26.

[30] A.-D. Marcu *et al.*, "Explainable artificial intelligence for energy-efficient radio resource management," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*. IEEE, 2023, pp. 1–6.

[31] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications: A new 5g technology for short-range vehicle-to-everything communications," *IEEE Veh. Technol. Mag.*, vol. 12, no. 4, pp. 30–39, 2017.

[32] J. Li *et al.*, "Effective joint scheduling and power allocation for URLLC-oriented V2I communications," *IEEE Trans. Veh. Technol.*, pp. 1–12, 2024.

[33] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[34] 3rd Generation Partnership Project, "Technical specification group radio access network; study on LTE-based V2X services; (release 14), document 3GPP TR 36.885 V14.0.0," 3rd Generation Partnership Project (3GPP), Technical Specification (TS), 2016. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.885

[35] K. Wang *et al.*, "Packet error probability and effective throughput for ultra-reliable and low-latency UAV communications," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 73–84, 2020.

[36] N. Khan and S. Coleri, "Resource allocation for ultra-reliable low-latency vehicular networks in finite blocklength regime," in *Proc. IEEE Int. Mediterranean Conf. Commun. Netw. (MeditCom)*, 2022, pp. 322–327.

[37] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[38] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[39] N. Khan and S. Coleri, "Event-triggered reinforcement learning based joint resource allocation for ultra-reliable low-latency V2X communications," *IEEE Trans. Veh. Technol.*, pp. 1–16, 2024.

[40] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural Inf. Process. Syst. 30*, pp. 4765–4774, 2017.

[41] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMlR, 2017, pp. 3145–3153.

[42] H. Yuan *et al.*, "An empirical study of the effect of background data size on the stability of shapley additive explanations (SHAP) for deep learning models," 2023. [Online]. Available: https://arxiv.org/abs/2204.11351

[43] P. Kyösti *et al.*, "Ist-4-027756 winner II D1.1.2 V1.2 winner II channel models," Feb 2007. [Online]. Available: https://www.cept.org/files/8339/winner2%20-%20final%20report.pdf

[44] C. She *et al.*, "Improving network availability of ultra-reliable and low-latency communications with multi-connectivity," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5482–5496, 2018.

[45] M. Hua *et al.*, "Multi-agent reinforcement learning for connected and automated vehicles control: Recent advancements and future prospects," 2024. [Online]. Available: https://arxiv.org/abs/2312.11084

[46] T. Rashid *et al.*, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 178, pp. 1–51, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-081.html

[47] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMLR, 2017, pp. 1126–1135.