

# Multi-Agent Deep Reinforcement Learning for Safe Autonomous Driving with RICS-Assisted MEC

Xueyao Zhang, *Student Member, IEEE*, Bo Yang, *Member, IEEE*, Xuelin Cao, *Member, IEEE*,

Zhiwen Yu, *Senior Member, IEEE*, George C. Alexandropoulos, *Senior Member, IEEE*,

Yan Zhang, *Fellow, IEEE*, Mérouane Debbah, *Fellow, IEEE*, and Chau Yuen, *Fellow, IEEE*

**Abstract**—Environment sensing and fusion via onboard sensors are envisioned to be widely applied in future autonomous driving networks. This paper considers a vehicular system with multiple self-driving vehicles that is assisted by multi-access edge computing (MEC), where image data collected by the sensors is offloaded from cellular vehicles to the MEC server using vehicle-to-infrastructure (V2I) links. Sensory data can also be shared among surrounding vehicles via vehicle-to-vehicle (V2V) communication links. To improve spectrum utilization, the V2V links may reuse the same frequency spectrum with V2I links, which may cause severe interference. To tackle this issue, we leverage reconfigurable intelligent computational surfaces (RICSs) to jointly enable V2I reflective links and mitigate interference appearing at the V2V links. Considering the limitations of traditional algorithms in addressing this problem, such as the assumption for quasi-static channel state information, which restricts their ability to adapt to dynamic environmental changes and leads to poor performance under frequently varying channel conditions, in this paper, we formulate the problem at hand as a Markov game. Our novel formulation is applied to time-varying channels subject to multi-user interference and introduces a collaborative learning mechanism among users. The considered optimization problem is solved via a driving safety-enabled multi-agent deep reinforcement learning (DS-MADRL) approach that capitalizes on the RICS presence. Our extensive numerical investigations showcase that the proposed reinforcement learning approach achieves faster convergence and significant enhancements in both data rate and driving safety, as compared to various state-of-the-art benchmarks.

**Index Terms**—Autonomous driving, reconfigurable intelligent computational surface, multi-access edge computing, multi-agent deep reinforcement learning.

## I. INTRODUCTION

X. Zhang and B. Yang are with the School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, 710129, China (email: yang\_bo@nwpu.edu.cn, 2024263006@mail.nwpu.edu.cn).

Z. Yu is with the School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, 710129, China, and Harbin Engineering University, Harbin, Heilongjiang, 150001, China (email: zhiwenyu@nwpu.edu.cn).

X. Cao is with the School of Cyber Engineering, Xidian University, Xi'an, Shaanxi, 710071, China (email: caoxuelin@xidian.edu.cn).

G. C. Alexandropoulos is with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 16122 Athens, Greece (email: alexandg@di.uoa.gr).

Y. Zhang is with the Department of Informatics, University of Oslo, 0316 Oslo, Norway (email: anzhang@ieee.org).

M. Debbah is with KU 6G Research Center, Department of Computer and Information Engineering, Khalifa University, Abu Dhabi 127788, UAE (email: merouane.debbah@ku.ac.ae)

C. Yuen is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore (email: chau.yuen@ntu.edu.sg).

WITH the rapid advancement of autonomous driving technology, the perception and decision-making capabilities of vehicles have become crucial for ensuring safe autonomous driving. However, autonomous vehicles (AVs) are required to process vast amounts of data collected from onboard sensors in real-time, imposing significant challenges to both computational resources and communication capabilities [1] [2]. This challenge is exacerbated in high-density vehicular ad hoc networks (VANETs) [3] [4], where latency-sensitive tasks —ranging from collision avoidance to real-time path planning— need to be executed within sub-second intervals. In this context, onboard devices are required to perform extensive sensing and computational tasks within a short time frame, while simultaneously relying on Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication links to transmit substantial volumes of data, all amidst the competition for limited spectral resources. Existing research has predominantly focused on enhancing modeling algorithms to improve computational accuracy or addressing real-world issues, such as path planning [5]. However, limited attention has been given to the interference on the same spectrum in high-density vehicular environments and the fulfillment of dynamic computational requirements, which are critical for ensuring both efficiency and safety.

By utilizing the dynamic modeling approach of Markov decision processes (MDPs), one can effectively capture the environmental conditions that change over time and make informed decisions in each time slot. This enables optimizing performance in rapidly evolving high-density vehicular network environments. To this end, in this paper, we adopt a DRL approach to effectively address the high coupling and complexity of the optimization variables. In particular, we present a novel autonomous driving network architecture based on the RICS technology and propose the driving safety-enabled multi-agent DRL (DS-MADRL) framework to address task offloading for AVs, spectrum sharing strategies, and joint optimization of the RICS parameters. It is demonstrated that our framework not only effectively enhances the spectrum efficiency and data rate of the considered system, but also significantly improves the real-time perception and safety of vehicles. Furthermore, compared to traditional algorithms, our proposed algorithm exhibits lower computational complexity and better adaptability.

The contributions of this paper are summarized below:

- A novel automated driving network system assisted by an RICS is presented. The system capitalizes on MEC to fa-

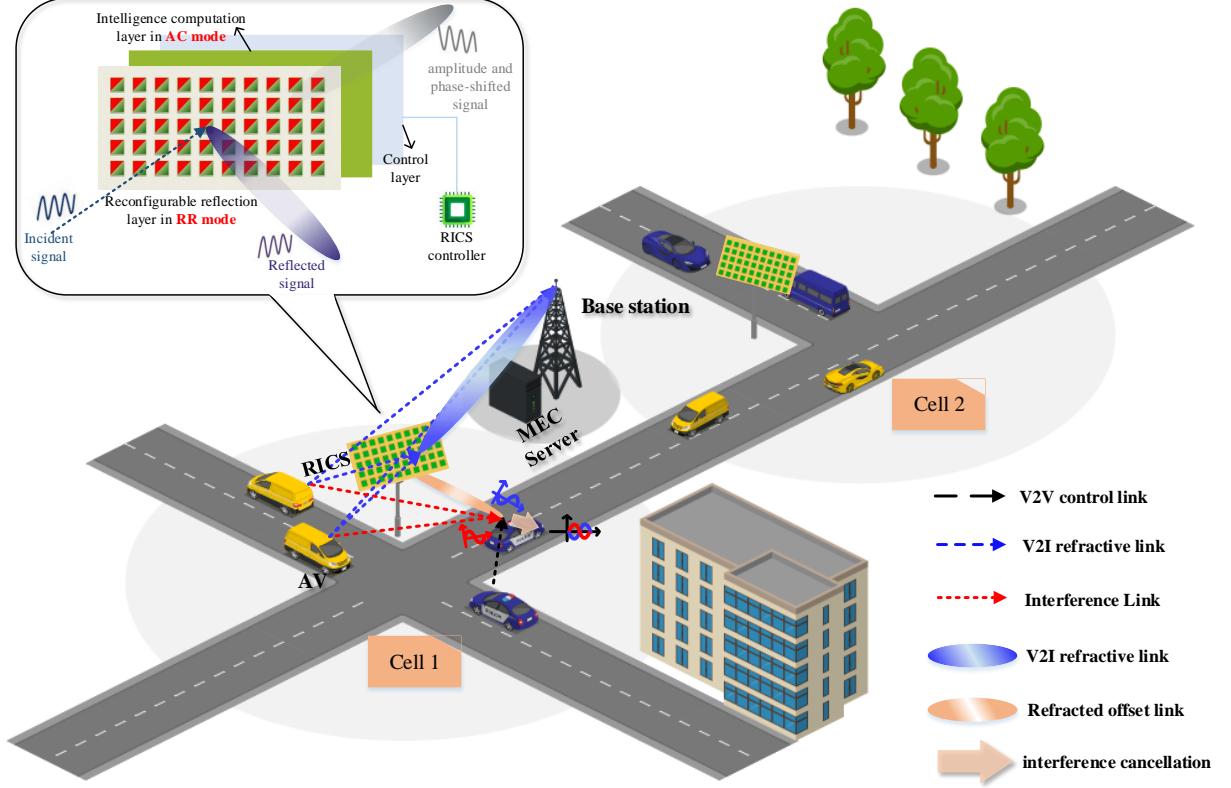


Fig. 1: The considered autonomous driving system model incorporating an RICS.

cilitate collaborative perception and decision-making between vehicles, aiming to address the limitations of computational resources and low communication efficiency faced by AVs. Our system model considers a multi-cell network architecture served by a single base station, where vehicles communicate with the BS via Vehicle-to-Infrastructure (V2I) links using Frequency Division Multiple Access (FDMA) for efficient channel allocation, thereby avoiding mutual interference. V2V communication enhances channel utilization by reusing AV channels, while local computational tasks can be partially offloaded to edge servers to alleviate local computational loads and reduce processing delays. RICS can simultaneously reflect and transmit signals to extend coverage while adjusting transmitted signals to mitigate interference introduced in V2V communication. We formulate a novel optimization problem incorporating parameters related to driving safety.

- We model our automated driving network optimization problem as a Markov game and solve it using a novel driving safety-enabled multi-agent deep reinforcement learning (DS-MADRL) framework. This framework innovatively combines hybrid action space optimization with safety-driven reward design. The proposed algorithm employs Q-decaying DQN (DDQN) to handle discrete actions and Multi-pass DQN (MP-DQN) networks, which

is simpler to train and more stable in discrete-continuous hybrid spaces, to address continuous-discrete actions. Additionally, the design of the reward function integrates the safety of AVs and the reliability of V2V communication to ensure system safety in dynamic environments. Moreover, we consider time-varying channels and interference among multiple users and introduce a cooperative learning mechanism among users. By designing a joint reward function to enhance system security, we effectively address the complex policy coordination challenges faced by two types of agents within a heterogeneous action space. The centralized training with decentralized execution (CTDE) framework is adopted to facilitate centralized training and distributed execution [38], thus, reducing the communication and computational overhead during online execution.

- The convergence performance of the proposed algorithmic framework is verified through extensive simulation results, ensuring the safety of autonomous driving. We also investigate the robustness of the algorithm in both single- and multi-cell scenarios and explore the role of amplitude adjustment coefficients in interference cancellation. Our extensive comparisons with traditional optimization algorithms, as well as DQN and deep deterministic policy gradients (DDPG) algorithms, showcase the superiority of the proposed approach.

TABLE I: System model Parameters

Parameter	Description
$\mathcal{C}$	Set of cells
$\mathcal{V}$	Set of V2Vs
$\mathcal{K}$	Set of elements of the RICS
$\beta_k^r$	Reflection amplitude of each element
$\beta_k^t$	Transmission amplitude of each element
$\Theta_r(l)$	The reflection coefficient matrices
$\Theta_t(l)$	The transmission coefficient matrices
$\mathbf{h}_{u,r} \in \mathbb{C}^{K \times 1}$	the channels from AV to RICS
$\mathbf{h}_{r,b} \in \mathbb{C}^{K \times 1}$	the channels from RICS to BS
$\mathbf{h}_{r,v} \in \mathbb{C}^{K \times 1}$	the channels from RICS to V2V
$\omega_{u,n}$	channel sharing
$\rho_{u,c}$	offloading ratio
$s_{u,c}$	The input data size for computation
$F_{u,c}$	Resources allocated to each AV by BS
$A_{u,c}$	The AVs' inference accuracy
$A_b$	The BS's inference accuracy

The paper is organized as follows: Section II introduces the RICS-assisted autonomous driving system model along with the channel model and our optimization problem formulation promoting driving safety. Section III presents the fundamentals of MADRL, models the optimization problem as a Markov game, and discusses the proposed DS-MADRL algorithm. Section VI verifies the proposed algorithm's effectiveness through simulation experiments and comparisons with state-of-the-art algorithms. The concluding remarks of the paper are included in Section V.

## II. RELATED WORKS

### A. MEC in Vehicular Networks

To address these challenges, Multi-Access Edge Computing (MEC) has been proposed as a solution to assist vehicles in offloading computation, while enhancing the efficiency of information transfer through V2I and V2V communications. This approach leverages the proximity of edge servers to reduce latency and improve reliability, making it particularly suitable for dynamic vehicular environments. In [6], a distributed multi-hop task offloading decision model is proposed to optimize task execution efficiency in MEC and V2I by leveraging vehicles with idle computational resources. In addition, [7] investigates novel task offloading algorithms aimed at optimizing offloading and resource allocation strategies [8]–[10]. However, these studies primarily focus on enhancing the offloading process, often assuming that interference from communication links is negligible, thus, failing to adequately account for the signal degradation caused by spectrum sharing [11]–[14]. This oversight complicates the ability of existing methods to effectively tackle signal degradation and fluctuations in communication quality in real-world dynamic traffic environments, which could potentially compromise the safety of autonomous driving.

### B. RIS for V2X Communications

Reconfigurable intelligent surfaces (RIS) have demonstrated significant potential in enhancing communication quality, such

as improving channel gain and reducing signal fading [15]–[20], while also offering advantages like low energy consumption and high energy efficiency [21], [22]. Recent advancements have seen the emergence of innovative RIS structures, such as filtered reconfigurable intelligent computational surface (RICS) [23], multi-layer RISs [24], hybrid simultaneous reflecting and sensing RISs [25], and simultaneous transmitting a reflecting (STAR) RISs [26], which are beginning to explore additional functionalities and capabilities. This exploration signifies a growing interest in leveraging RIS technology within the field of vehicular networks, with recent studies focusing on resource allocation, communication reliability, channel estimation, as well as performance analysis in RIS-assisted V2X systems [27], [28]. However, within the context of high-density vehicular networks, the unilateral reflective capability of conventional RISs limits severe signal coverage while being highly impacted by interference conditions. The innovative structure of an RICS in [29] has been shown to be configured in various ways to adapt to different scenarios. It is designed to simultaneously transmit and reflect signals, thereby improving coverage and adapting the V2I refraction link to mitigate interference on the V2V link. RICS optimizes the quality of wireless communications and adaptively modifies signals due to its computational ability. This capability enhances perception and decision-making efficiency for automated driving in complex dynamic environments.

### C. DRL Foundations

The evolution of deep reinforcement learning has fundamentally transformed optimization in complex dynamic systems. Building upon the foundational deep Q-network (DQN) that integrated experience replay and target networks for stable value estimation, subsequent innovations addressed critical limitations in real-world deployment. Decaying DQN (DDQN) [34] employs  $\varepsilon$ -greed to adaptively balance exploration-exploitation tradeoffs, ensuring smooth policy convergence in dynamic networks with time-varying environments. For continuous control, Deep Deterministic Policy Gradient (DDPG) [35] combined actor-critic architectures with off-policy learning to efficiently optimize policies. The evolution of hybrid action space optimization saw critical breakthroughs with Parameterized DQN (P-DQN) [36], which decoupled discrete and continuous policy networks for joint action learning, yet faced gradient interference between decision dimensions. Multi-Pass DQN (MP-DQN) [37] resolved this by isolating action-specific gradients through masked basis vector propagation, enabling stable training in complex decision spaces. In multi-agent scenarios, Centralized Training with Decentralized Execution (CTDE) [49] emerged as a foundational paradigm, allowing collaborative policy learning while preserving decentralized execution efficiency—a crucial balance for systems requiring both coordination and operational autonomy.

### D. DRL for Communication Systems

Despite the latter advancements, the integration of RICS into autonomous driving networks introduces a multi-dimensional optimization challenge. Typically, traditional optimization methods, such as Lagrange multipliers [30], KKT

conditions [31], manifold optimization (MO) [32], dyadic optimization, and alternating optimization [33], are commonly employed to tackle the modeled non-convex problems. However, these problems are often NP-hard, and while convergence may be achievable in static scenarios, this approach falls short when addressing the dynamic nature of real-world conditions, particularly due to the significant computational overhead involved in recalculating solutions as environmental conditions change over time. In high-density vehicular networks, the optimization variables are highly coupled, and the phase shift selection for RICS becomes increasingly complex due to varying conditions at different locations and times, as well as differing channel states. To overcome these limitations, some studies have considered integrating deep reinforcement learning (DRL) into the optimization of wireless communications [39]–[43]. The research in [44] delves into the use of DRL to optimize drone trajectories and beamforming. Additionally, [45] studies the problem of the enhancement of V2I communication using an RIS in vehicular edge computing systems and addresses optimization problems through reinforcement learning methods. Besides, [46] explores a STAR-RIS-assisted V2X communication system that combines STAR-RIS with DRL algorithms. Additionally, [47] utilized DRL with Lyapunov optimization to investigate the trade-offs among computation, communication, and latency in RIS-enabled MEC systems. Building upon this literature, [48] further employed a scalable multi-agent DRL (MADRL) framework for optimizing user scheduling and precoding in distributed RIS-aided communication systems. However, little progress has been made in applying DRL for RIS-integrated time-varying V2V/V2I systems.

### III. RICS-ASSISTED AV SYSTEMS

In this section, we present the RICS-assisted autonomous driving system model and the channel model used in this paper. We also introduce the computation model for the RICS and our design optimization formulation.

#### A. Network Model

We investigate an uplink autonomous driving scenario facilitated by RICS, as illustrated in Fig. 1. This scenario consists of a base station (BS) that simultaneously serves multiple cells, which can be denoted as  $\mathcal{C} = \{1, 2, \dots, C\}$ . Each cell is equipped with a RICS that has  $K$  reflecting elements and  $A$  AVs that communicate with the BS via Vehicle-to-Infrastructure (V2I) links, while also sharing information with several Vehicle-to-Vehicle (V2V) pairs. The sets of AVs and V2Vs can be represented as follows:  $\mathcal{U} = \{1, 2, \dots, U\}$  and  $\mathcal{V} = \{1, 2, \dots, V\}$ . In this system, AVs have the capability to transmit computing tasks to the MEC server through V2I links, as well as share information with other vehicles using V2V links.

To accommodate the simultaneous information transmission needs of multiple vehicles to the BS for computational assistance, a Frequency Division Multiple Access (FDMA) approach is employed [50]. Each AV uses its allocated sub-channel for information transmission, thereby avoiding mutual interference. The total bandwidth is denoted as  $W$ , and the  $U$

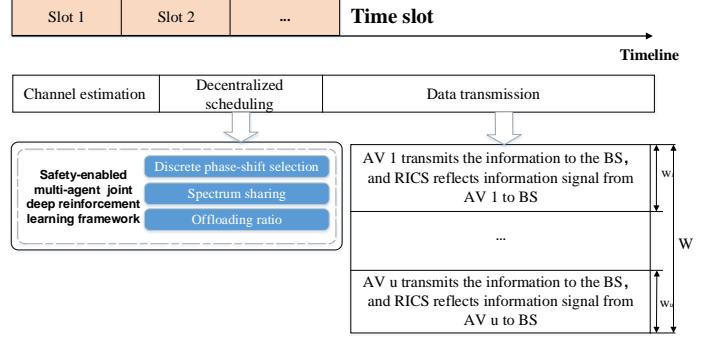


Fig. 2: The proposed DS-MADRL mechanism.

AVs share the channel equally. We assume that the total transmission and computation time for AVs across multiple cells is divided into  $L$  equal, non-overlapping time slots. During each time slot, AVs occupy their sub-channels following the FDMA protocol. To enhance spectrum utilization, V2V pairs may share the sub-channel of a specific AV for information transmission. Spectrum sharing is represented by a binary variable  $\omega_{u,v}$ , where  $\omega_{u,v}$  indicates channel sharing between the  $u$ -th AV and the  $v$ -th V2V, causing interference to the V2V pairs. In each time slot, we assume that both the AVs and the V2Vs remain stationary. After completing the algorithm during this time slot, they will move to their positions in the subsequent  $(l+1)$ -th time slot.

Furthermore, The specific structural configuration of the RICS is referenced in [51]. To put it simply, the first layer operates in reflection-refraction (RR) mode, which reflects a portion of the incident signal, with the reset facilitating signal refraction to support V2V users on the opposite side, thus extending the signal coverage. The second layer works in AC mode, utilizing metamaterials to adjust the amplitude of the transmitted signals, thereby generating signals that can interfere destructively with V2I interference waves to achieve interference cancellation. Specifically, the energy splitting ratio on the reconfigurable intelligent surface is defined as  $\chi_k \triangleq \beta_k^r : \beta_k^t$ ,  $\forall k \in \mathcal{K} \triangleq \{1, 2, \dots, K\}$ , where  $\beta_k^r$  and  $\beta_k^t$  both in  $[0, 1]$  and  $\beta_k^r + \beta_k^t = 1$ . The amplitude adjustment factor for the  $k$ -th element of the intelligent computing layer on transmitted signals is denoted as  $\Psi_k$ . The reflection-refraction signals can then be expressed as:  $s_k^r(l) = \sqrt{\beta_k^r} e^{j\theta_k^r(l)} s_k$  and  $s_k^t(l) = \sqrt{\beta_k^t} e^{j\theta_k^t(l)} s_k$ , with  $\theta_k^r(l), \theta_k^t(l) \in (0, 2\pi]$ . The refraction-reflection coefficient matrix for the  $i$ -th RICS are given by  $\Theta_r(l) = \text{diag}\{s_1^r(l), s_2^r(l), \dots, s_K^r(l)\}$  and  $\Theta_t(l) = \text{diag}\{s_1^t(l), s_2^t(l), \dots, s_K^t(l)\}$ , respectively.

#### B. Channel Model

In the proposed  $c$ -th cell of the RICS-assisted autonomous driving system, we define  $\mathbf{h}_{u,r} \in \mathbb{C}^{K \times 1}$ ,  $\mathbf{h}_{r,b} \in \mathbb{C}^{1 \times K}$ , and  $\mathbf{h}_{r,v} \in \mathbb{C}^{K \times 1}$  to represent the channels from the  $u$ -th AV to the RICS, from the RICS to the BS, and from the RICS to the  $v$ -th V2V pair, respectively. The modeling of wireless channels takes into account both large-scale and small-scale

fading components. Large-scale fading is primarily determined by distance, which can be represented by  $P(l) = \sqrt{C_0 \left( \frac{d(l)}{d_0} \right)^\alpha}$  [52]. Small-scale fading, on the other hand, is influenced by the environment, including the line-of-sight and Non-line-of-sight (NLoS). We denote  $\mathbf{h}^{Los}$  to represent the direct link, which is closely related to the placement angles of the antennas. In this system, the RICS adopts a Uniform Linear Array (ULA) configuration, and therefore the LoS channel can be expressed as  $\mathbf{h}^{Los} = \alpha_T \alpha_R$ , where  $\alpha_t$  and  $\alpha_r$  represent the angular vectors and the transmitting and receiving antennas. NLoS consists of multiple reflected paths superimposed upon the direct link, typically following a circular complex Gaussian distribution, i.e.  $\mathbf{h}^{NLoS} \sim \mathcal{CN}(0, 1)$ . Thus, the channel gains can be summarized as follows

$$\mathbf{h}_{u,r}(l) = P(l) \left( \sqrt{\frac{\zeta_{u,r}}{1 + \zeta_{u,r}}} \mathbf{h}_{u,r}^{Los} + \sqrt{\frac{1}{1 + \zeta_{u,r}}} \mathbf{h}_{u,r}^{NLoS} \right), \quad (1)$$

$$\mathbf{h}_{r,v}(l) = P(l) \left( \sqrt{\frac{\zeta_{r,v}}{1 + \zeta_{r,v}}} \mathbf{h}_{r,v}^{Los} + \sqrt{\frac{1}{1 + \zeta_{r,v}}} \mathbf{h}_{r,v}^{NLoS} \right). \quad (2)$$

where,  $\zeta_{u,r}, \zeta_{r,v}$  are denoted as Rayleigh factors in small-scale fading. In addition, we define the channel of the direct link as:  $h_{u,b}$  and  $h_v$ , which denote the channel gains from the  $u$ -th AV to the BS and from the transmitter to the receiver of the  $v$ -th V2V pair, respectively. The interference link has  $h_{u,v}$  and  $h_{v,b}$  denoting the interference channel gains from the  $u$ -th AV to the Rx of the  $v$ -th V2V pair, respectively. We assume that both BS and vehicles have perfect channel state information (CSI), which will be preprocessed at the beginning of each time slot for channel estimation.

Based on the above analysis, the SINR received for the  $u$ -th AV and the  $v$ -th V2V pair can be derived as (3)(4), where  $P_u$  and  $P_t$  represent the transmission power of the AV and the Tx of the V2V pair, respectively. Therefore, the achievable uplink rate for the  $u$ -th AV and  $v$ -th V2V is given by (5)(6). Similarly, the achievable data rate for the  $v$ -th V2V pair can be obtained.

$$\gamma_b^u(l) = \frac{P_u(l) |h_{u,b}(l) + \mathbf{h}_{r,b}(l) \Theta_r(l) \mathbf{h}_{u,r}(l)|^2}{\sum_{v=1}^V \omega_{u,v}(l) P_t(l) |\mathbf{h}_{v,b}(l)|^2 + W \xi_0}, \quad (3)$$

$$\gamma_v(l) = \frac{P_t(l) |h_v(l)|^2}{\sum_{u=1}^U \omega_{u,v}(l) P_u(l) |h_{u,v}(l) + \mathbf{h}_{r,v}^H(l) \Theta_r(l) \mathbf{h}_{u,r}(l)|^2 + W \xi_0}, \quad (4)$$

$$R_b^u(t) = \frac{W}{u} \log_2(1 + \gamma_b^u(l)). \quad (5)$$

$$R^v(t) = \frac{W}{v} \log_2(1 + \gamma_v(l)). \quad (6)$$

### C. RICS Computation Model

In RICS-assisted autonomous driving networks, real-time sensor fusion and computation offloading are crucial for

ensuring driving safety. Sensor fusion integrates data from multiple sensors to provide a more comprehensive and accurate understanding of the vehicle's surrounding environment. Each vehicle is required to perform a series of perception and decision-making tasks. Furthermore, the proposed partial offloading model allows vehicles to offload part of the task to the MEC server. This approach reduces the CPU usage of onboard devices, thereby lowering energy consumption and improving system response times to ensure safe driving.

To simplify the model, we describe the tasks executed by the AVs using three variables:  $s_{u,c}$ [bits] represents input data size for computation,  $\varsigma_{u,c}$  [cycles] indicates the CPU cycles required to process  $s_{u,c}$ , and  $\sigma_{u,c}$  [secs] denotes the maximum allowable delay. To improve computational efficiency, the models deployed at BS, and the vehicles are configured according to their hardware capabilities, ensuring fast response times and high-precision environmental perception. When processing image data of a specific quality  $q$ , the AVs DNN-based inference accuracy is guaranteed no larger than the BS's inference accuracy, i.e.,  $A_{u,c}(q) = \lambda A_b(q)$ ,  $0 \leq A_{u,c}(q), A_b(q), \lambda \leq 1$ .

After providing the above definitions, a partial offloading model is designed to fully utilize computational resources. Specifically, the offloading ratio  $\rho_u$  represents the ratio of tasks offloaded to the BS, with  $1 - \rho_u$  indicating the data ratio that needs to be processed locally. It is important to note that the tasks being partially offloaded by vehicles are divisible video sequence tasks. This is because a video sequence can be decomposed into multiple independent frames or segments, each of which can be processed independently [53]. Therefore, each task on an AV can be divided into two parts: the local computation delay of  $u$ -th AV denoted as

$$\tau_{loc}^{u,c}(l) = (1 - \rho_{u,c}(l)) \frac{s_{u,c}}{f_{u,c}}. \quad (7)$$

Considering the scenario where multiple cells share a single BS, the delay in computation offloading considers that multiple cells may simultaneously upload tasks to the BS for processing. In this case, a resource allocation strategy is employed where the BS evenly distributes computational resources among multiple cells. Since the tasks from each cell are processed in parallel, the overall computation delay is determined by the cell that requires the most time to complete. Thus, the computation offloading delay can be expressed as follows:

$$T_{off}^{u,c}(l) = \rho_{u,c}(l) \left( \frac{s_{u,c}}{R_b^{u,c}(l)} + \max \left( \frac{\varsigma_{u,c}}{F_{u,c}} \right) \right), \quad (8)$$

where  $f_{u,c}$  represents the computational resources of the AVs, while  $F_{u,c}$  indicates the resources allocated by the BS for the computational tasks assigned to each AV. Here, the BS distributes the task resources evenly among all vehicles. As a result, the total delay of a task on the  $u$ -th AV is calculated as  $\tau_{u,c}(l) = \max\{\tau_{loc}^{u,c}(l), \tau_{off}^{u,c}(l)\}$ . Based on this, the average inference accuracy of tasks can be obtained

$$\tilde{A}_{u,c}(q) = (1 - \rho_{u,c}(l)) A_{u,c}(q) + \rho_{u,c}(l) A_b(q). \quad (9)$$

#### D. Problem Formulation

Using the latter expressions, We define the driving safety factor as follows:

$$S_{u,c}(l) = \frac{\tilde{A}_{u,c}(q)}{\tau_{u,c}} = \frac{(1 - \rho_{u,c}) A_{u,c}(q) + \rho_{u,c} A_b(q)}{\max\{\tau_{loc}^{u,c}, \tau_{off}^{u,c}\}}. \quad (10)$$

Our goal is to maximize the sum safety factor of the AVs while satisfying the outage probability of V2V pairs, i.e., in mathematical terms:

$$\mathbb{P}: \max_{\omega, \Theta_x, \rho} \frac{1}{L} \sum_l^L \sum_c^C \sum_u^U S_{u,c}(l) \quad (11a)$$

$$\text{s.t. } \Pr\{\gamma_v(l) \leq \gamma_{th}\} \leq P_{outage}(l), \forall l, \quad (11a)$$

$$\omega_{u,v}(l) \in 0, 1, \quad \forall u, \forall v, \forall l, \quad (11b)$$

$$\sum_{v=1}^V \omega_{u,v}(l) \leq 1, \quad \forall u, \forall v, \forall l, \quad (11c)$$

$$\beta_k^r + \beta_k^t = 1, \quad 1 \leq k \leq K, \quad (11d)$$

$$\rho_u(l) \in [0, 1], \quad \forall u, \quad (11e)$$

where  $\omega = \{\omega_{u,v}, \forall u, v\}$  and  $\rho = \{\rho_1, \rho_2, \dots, \rho_u\}$ .

#### E. Outage Probability of V2V Pairs

In this section, we address the outage probability constraint of V2V pairs in (11a) by approximating it using a smooth step function  $\hat{u}_\delta(x) = \frac{1}{1+e^{-\delta x}}$  [54], which includes a smoothing parameter  $\delta$  to control the approximation error. Subsequently, we can approximate the constraint (11a):

$$\mathbb{E}[\hat{u}_\delta(\gamma_{th} - \gamma_v(l))] \leq P_{outage}(l). \quad (12)$$

With the approximated constraint provided in (12), we can calculate the outage probability in constraint (11a), leading to a further simplification, as outlined in **Theorem 1**.

**Theorem 1.** Let  $\tilde{\gamma}_v(\omega(l), \Theta_x(l), \rho(l)) = \mathbb{E}[\gamma_v]$ , we can represent the constraint (11a) as

$$\tilde{\gamma}_v(\omega(l), \Theta_x(l), \rho(l)) \geq \gamma_{th} + \frac{1}{\varpi} \ln \left( \frac{1}{P_{outage}(l)} - 1 \right) \triangleq \tilde{\gamma}_c(l). \quad (13)$$

*Proof.* Please refer to [51].  $\square$

Based on **Theorem 1**, we can conveniently reformulate constraint (11a) to address the optimization problem  $\mathbb{P}$ , leading to the following equivalent problem:

$$\mathbb{P}: \max_{\omega, \Theta_x, \rho} \frac{1}{L} \sum_l^L \sum_c^C \sum_u^U S_{u,c}(l) \quad (14)$$

s.t. (11b) – (11e), (13).

Note that this problem is non-convex according to [51]. Additionally, a strong coupling among the three optimization variables complicates the problem solution further. Traditional algorithms often use alternating optimization techniques to approach the optimal solution; however, these methods struggle to handle time-varying channels. Motivated by this limitation, we develop a DS-MADRL algorithm to find a viable solution for  $\mathbb{P}$ .

#### IV. RICS-ASSISTED DRIVING SAFETY MAXIMIZATION

In this section, the proposed DS-MADRL scheme, its training, and algorithmic steps are introduced. The section commences with a brief description of MADRL principles.

##### A. Markov Game Formula

RL is an important branch of ML, which learns the optimal strategy to maximize long-term reward through the interaction between the agent and the environment. DRL integrates the decision planning ability of RL and the feature learning ability of DL, so that the agent can process high-dimensional and complex input data and learn more complex strategies [55].

In the previous section, we modeled the optimization problem (14) as a continuous decision process over a time frame, i.e, the next time slot will make the decision based on the current state. The agent aims to maximize the long-run expected discount reward by learning an optimal strategy  $P$ . This learning is based on interaction with the environment, which is modeled as a Markov decision process (MDP) denoted by a quintuple  $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$ . This representation encompasses the state space, the action space, the reward function, the transport strategy, and the discount factor. However, in the complex scenario of this paper, a single agent may be affected by other individuals, and it is difficult to learn the overall strategy. Therefore, we introduce multiple agents here, and the corresponding Markov process also becomes the Markov game (MG). In MG, multiple agents interact in a shared environment, and each agent can choose its own actions to affect the environment. In contrast to MDP, MG considers the competition and cooperation between multiple agents. Correspondingly, MG can also be described by a quintuple:  $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ , where  $\mathcal{N}$  is the number of agents involved and  $\mathcal{A}$  is the Cartesian product of all agents' action spaces.  $\mathcal{P}$  denotes the transition probability from one state to another, and  $\mathcal{R}$  indicates the reward function of each agent, that is, the immediate reward function obtained by each agent under a given state and action. The long-term discounted reward in the multi-agent setting MG represents the goal that each agent focuses on when selecting actions. This reward takes into account future uncertainty and discounts future rewards to ensure that the impact of distant rewards on the behavior of agents is not overly overlooked.

In MG, each agent selects actions that maximize its expected long-term cumulative discounted reward. By considering the impact of discount factors and future rewards, the agent can strike a better balance between immediate and long-term expected rewards, leading to more astute decisions and action selections. To achieve the maximum long-term cumulative discounted reward, in time slot  $l$ , our long-term cumulative discounted reward is given by:

$$G_l = \sum_{i=0}^L \gamma^i R_{l+i}. \quad (15)$$

In this context, the discount factor  $\gamma \in [0, 1]$  signifies the importance of future rewards to the agent. A higher value of  $\gamma$  near 1 indicates that the agent places greater emphasis on

long-term cumulative rewards. Conversely, a lower value of  $\gamma$  closer to 0 suggests that the agent prioritizes immediate rewards.

Based on the above model, the MG elements of the RICS-assisted autonomous driving vehicular network system are described as follows.

- 1) *State space:* The state space is the collection of system information at a specific time point, used to guide decision-making. It consists of the local states of RICS and AVs. For the RICS, the state space is defined by considering the historical actions taken by the system along with their corresponding channel states, which provide context for future decisions. For the AVs, the state is characterized by the current power level of each vehicle and the prevailing channel state, which reflects the vehicle's operational status and its ability to communicate effectively with the RIS and other entities in the network. The specific modeling of the state space for both the RIS and AVs is detailed as follows

$$s_{RICS} \triangleq [a_k(l-1), \mathbf{h}_{u,r}(l), \mathbf{h}_{r,v}(l), \mathbf{h}_{r,b}(l)], \quad (16)$$

$$s_u \triangleq [P_u(l), P_l(l), a_u(l-1), h_{u,v}(l), h_{u,b}(l)]. \quad (17)$$

The global state set composed of the local states of all agents is defined as:

$$S_l \triangleq [s_{RICS}, s_u]. \quad (18)$$

- 2) *Action space:* The action space mainly describes the set of all actions that the system can take. An action is the course of action taken by the system at a specific time point, used to alter its state. Due to the specific nature of the RICS structure, the amplitude coefficients of its refraction-reflection coefficient matrix and the amplitude adjustment factor of AC mode are limited by the material properties and cannot be adjusted over time. Therefore, we do not optimize them, and for the amplitude coefficients, we adopt a balanced transmission and reflection mode, i.e.,  $\beta_t = \beta_r = 0.5$ , evenly distributing energy between transmission and reflection. In addition, if each element of RICS is independently controlled, a large number of parameters are required, which would increase the training overhead. To address this issue, we partition the RICS into Q sub-blocks, assigning the same phase shift values to the elements within each sub-block. The feasible domain of discrete phase shift adjustments for the sub-blocks is as follows

$$\theta_l^t, \theta_l^r \in \left\{ 0, \frac{2\pi}{2^h}, \dots, \frac{2\pi(2^h - 1)}{2^h} \right\}, \quad (19)$$

$h$  represents the resolution bits. The corresponding RICS action space is defined as follows:

$$a_{RICS} \triangleq [\{\theta_l^t, \theta_l^r\}]. \quad (20)$$

The local action spectrum sharing strategy  $\omega$  of AVs is a discrete action, corresponding to (11b). The offload ratio  $\rho$  is a continuous action, corresponding to (11e).

define their action space as:

$$a_u \triangleq [\omega_u, \rho_u]. \quad (21)$$

Therefore, the local actions of all agents constitute joint actions:

$$\mathcal{A}_l \triangleq [a_{RICS}, a_u]. \quad (22)$$

- 3) *Reward:* The reward function is the immediate reward obtained after the system takes a specific action. Based on the optimization problem (14), we define it as:

$$r \triangleq \underbrace{\sum_{u=1}^U S_u}_{part1} + \underbrace{\min_{v=1}^V \{\tilde{\gamma}_v(\omega, \Theta_x) - \tilde{\gamma}_c, 0\}}_{part2}. \quad (23)$$

The reward function that guides the learning process is consistent with the objectives of multi-objective optimization. To achieve the goal of maximizing the total safety index of AVs, subject to the constraints (11c), we introduce a penalty, and if any of these constraints are not satisfied, the constraint set is terminated.

$$R(l) = \begin{cases} -Penalty, & \text{if } S_u = NS \\ r(l), & \text{otherwise,} \end{cases} \quad (24)$$

where  $NS$  represents a negative state, indicating that when the state of a certain AV fails to satisfy the constraint (11c). The state here includes the actions taken by the agent in the previous stage, and if the action violates the constraints, a penalty is imposed. In other words, the algorithm receives a negative reward  $Penalty$ , where  $Penalty$  is a sufficiently large positive number used to ensure that the algorithm quickly identifies and avoids invalid paths when encountering unsatisfied conditions, thereby improving its convergence speed.

## B. DS-MADRL Network Architecture

For the RICS-assisted autonomous driving vehicular system model presented in this paper, we design DS-MADRL to accommodate two types of agents. This framework aims to integrate the action selection tasks of various types of agents through joint decision-making and interaction with the environment, ultimately enhancing system safety. Considering the different types of actions performed by the two kinds of agents, we incorporate DDQN and MP-DQN networks to address the discrete phase shift selection problem of RICS, as well as the spectrum sharing strategy and task offloading ratio. The introduction of DDQN is specifically intended to tackle the discrete phase shift selection issue for RICS. Since the phase shifts for RICS are discrete, DDQN learns the mapping relationship between phase shifts and channel conditions, enabling it to select the optimal phase shift values to enhance data transmission rates. On the other hand, since AVs must simultaneously manage the task offloading ratios and spectrum-sharing strategies, we model this decision-making process as a mixed continuous-discrete action space. The MP-DQN network is well-suited to effectively learn in such hybrid action spaces, ensuring the maximization of system

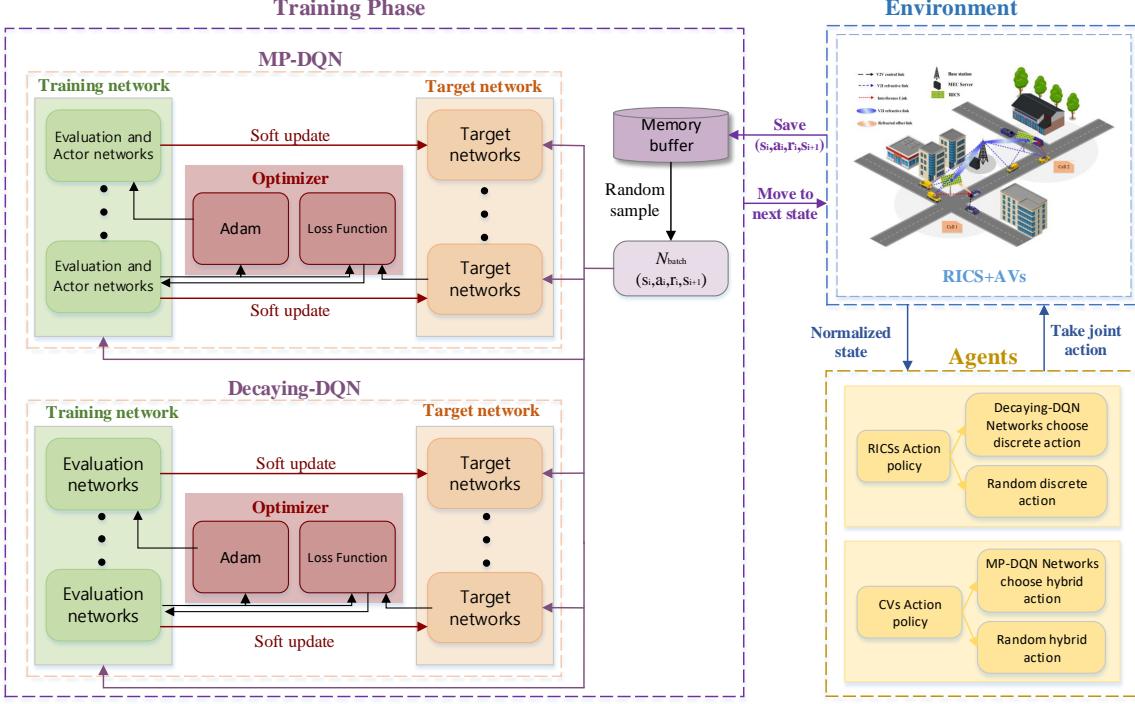


Fig. 3: The proposed training framework for our DS-MADRL algorithm.

safety across different environments. We anticipate that this joint learning framework will facilitate effective collaborative decision-making in complex interactions and dynamic environments, thereby improving the overall system performance.

Deep Q-Network (DQN) is a reinforcement learning algorithm that combines Q-learning and deep learning, aiming at solving decision-making problems in discrete high-dimensional state spaces. DQN uses a neural network to estimate the state-action value function, i.e.,  $Q(s, a) = \mathbb{E}_\pi [G_t | s_t = s, a_t = a]$ , which we call the Q value. The goal is to find the optimal strategy  $\pi^* = \arg \max_a Q^*(s, a)$  to maximize this Q value. The DQN structure includes a evaluation Q network to estimate the state-action value function, which maximizes  $Q(s, a)$  by selecting action  $a$ , which we define as:  $Q^*(s(l), a(l); \delta_w) = \mathbb{E} [R(l) + \gamma \max_{a(l+1)} Q(s(l+1), a(l+1))]$ , where  $\delta$  represents the parameter of the Q network. And a target Q network with the same structure as it is used to calculate the target  $Q^*(s, a)$  value. During the training process, the parameters of the  $Q_{tar}$  network are not immediately updated at each training step, but instead, the update is delayed for a certain period. These two neural networks are updated by minimizing the loss function, as shown in equation (25), and the evaluation of the state-action values is updated using the Bellman equation, as described by  $Q(s(l), a(l)) = \mathbb{E} [R(l) + \gamma \max_{a(l+1)} Q_{tar}(s(l+1), a(l+1))]$ . Through the evaluation of the Q-network, the appropriate strategy for DQN is to select a policy that maximizes the state-action values. In order to make the training more stable and quickly converge to the optimal strategy, we introduce an improved

DQN algorithm, decaying DQN, which introduces a Q-decay mechanism in the training process. It can balance accelerated convergence and avoid falling into the local optimal solution during the training process. The formula for the decaying learning rate is expressed as follows:

$$\alpha(episodes) = \alpha_0 (1 / (1 + \varepsilon \times episodes)), \quad (27)$$

where  $\alpha$  denotes the learning rate, a variable related to the number of training episodes,  $\alpha_0$  denotes the initial learning rate, and  $\varepsilon$  denotes the learning rate decay.

Parametrized DQN (P-DQN) is a network that can handle continuous and discrete action spaces and consists of two main networks: the Q network and the actor-network. Specifically, the Q network is used to receive the state and the joint action parameters  $s, x$  as inputs, and outputs the Q-value corresponding to each discrete action. In contrast, the actor-network receives the state as an input and outputs the best continuous action corresponding to each discrete action. However, this network suffers from a critical problem: it inputs all action parameters jointly into the Q network, which may result in each Q value  $Q_i$  being a function of all action parameters  $x$ , and not just the continuous action parameter  $x_i$  associated with that discrete action, which can lead to problems of ineffective gradients and suboptimal action selection. The MP-DQN solves this problem by separating action parameters through multiple passes to separate the action parameters.

In MP-DQN, the Bellman equation is redefined as (26) to accommodate cases with both continuous and discrete action spaces. For each discrete action  $a$ ,  $x_a^*$  is obtained by calculating  $x_a^* = \arg \sup_{x_{a(l+1)}} Q(s(l+1), a(l+1), x_a^*)$ ,

$$\mathcal{L}(\delta_w) = (R(l) + \tau \max_{a(l+1)} Q(s(l+1), a(l+1); \delta_w^-) - Q(s(l), a(l); \delta_w))^2 \quad (25)$$

$$Q(s(l), a(l), x_a(l+1)) = \mathbb{E} \left[ R(l) + \gamma \max_{a(l+1)} \sup_{x_a(l+1)} Q(s(l+1), a(l+1), x_a(l+1)) \right] \quad (26)$$


---

**Algorithm 1:** DS-MADRL Training for  $\mathbb{P}$  in (14)

**Result:** Sum safety factor  $S_u$ , sum V2V data rate  $R_v$ , offloading policy  $\phi_p$ , spectrum sharing policy  $\phi_w$ , and refraction reflection coefficient matrix policy  $\phi_\theta$

```

1 . Initialize all DDQN agents networks  $Q(s, a; \delta_w)$  and
    $Q(s, a; \delta_w^-)$  with  $\delta_w = \delta_w^-$ , and also initialize all MP-DQN
   agents' networks  $Q(s, a, x_a; \delta_q)$ ,  $x_a(s; \delta_x)$ ,
    $Q(s, a, x_a; \delta_q^-)$ ,  $x_a(s; \delta_x^-)$  with  $\delta_q = \delta_q^-$  and  $\delta_x = \delta_x^-$ ;
2 for  $e = 1, 2, \dots, E$  do
3   Reset vehicle positions and generate initial state  $s(0)$ ;
4   for  $t = 1, 2, \dots, T$  do
5     for each RICS sub-blocks  $q = 1, \dots, Q$  do
6       Each  $q$ -th DDQN agent selects a discrete phase
       shift  $a_2$  using  $\epsilon$ -greedy algorithm;
7     end
8     for each AV  $m = 1, 2, \dots, M$  do
9       Each  $u$ -th MP-DQN agent selects a joint action
        $a_1$  using  $\epsilon$ -greedy policy;
10    end
11    The environment executes joint action  $a(t)$ , obtains
       the reward  $r(t)$  based on (24) and transitions to the
       next state;
12    Store tuple  $(s(t), a(t), r(t), s(t+1))$  in the
       experience replay buffer;
13    Sample a mini-batch of size  $B$  from the experience
       replay buffer;
14    Update weights  $\delta_w$  for all DDQN agents by
       minimizing the loss function according to (25) and
       update weights  $\delta^-$  by copying  $\delta_w$ ;
15    Update weights  $\delta_q^-$  and  $\delta_x^-$  for all MP-DQN
       networks based on (30)(29);
16    Additionally, update the target networks using a soft
       replacement approach;
17  end
18 end

```

---

and then  $a^*$  is obtained by calculate  $a^* = \arg \max_{a(l+1)} Q(s(l+1), a(l+1), x_a^*)$ . The MP-DQN structure uses deterministic policy network  $x_a(s; \delta_x)$  to approximate continuous action  $x_a^Q = \arg \sup_{x_a} Q(s, a, x_a)$ , and the evaluation Q network  $Q(s, a, x_a; \delta_q)$  is used to approximate the state-action value function  $Q(s, a, x_a)$ . In addition, there is a corresponding target  $Q(s, a, x_a; \delta_q^-)$  network and target policy network  $x_a(s; \delta_x^-)$ . MP-DQN works similarly to DQN by interacting with the environment and storing experiences in the experience replay buffer. When training, it randomly selects a set of tuples from the experience replay buffer to train to reduce the correlation between observations and decisions. In addition, the target Q value of MP-DQN at the  $l$  training step is as follows:

$$y(l) = R(l) + \tau \max Q(s(l+1), a, x_a(s(l+1); \delta_x^-); \delta_q^-). \quad (28)$$

Next, we update the parameters of the Q network and the policy network by minimizing the loss function, specifying

the loss function of the deterministic policy network, and evaluating the Q-network given by

$$\mathcal{L}(\delta_x) = - \sum_{a(l) \in \mathcal{A}} Q(s(l), a(l), x_a(s(l); \delta_x); \delta_q(l)), \quad (29)$$

$$\mathcal{L}(\delta_q) = \frac{1}{2}(y(l) - Q(s(l), a(l), x_a(l); \delta_q))^2. \quad (30)$$

Based on the (29)(30) above, the gradient of Q-values estimated by the Q network is backpropagated to the actor-network, which guides it to learn the optimal action parameter policy. To address the issue of joint action parameter input present in the P-DQN framework, MP-DQN introduces a standard basis vector  $x_e^a$ , where only the  $a$ -th action parameter  $x_a$  is non-zero, while all other action parameters remain zero. As a result, the Q network computes the Q-value solely based on the current action parameter, effectively eliminating the influence of invalid gradients. MP-DQN uses a multi-channel approach that allows small batches of data containing  $B$  tuples of  $|a|$  actions to be processed in the same way as small batches of size  $|a|$ :

$$\begin{pmatrix} Q(s, a, xe_1) \\ Q(s, a, xe_2) \\ \vdots \\ Q(s, a, xe_{|a|}) \end{pmatrix} = \begin{pmatrix} Q_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{|a||a|} \end{pmatrix}, \quad (31)$$

where,  $Q_{i,j}$  represents the Q-value for action  $j$  computed during the  $i$ -th path. It's important to note that only the diagonal elements  $Q_{i,i}$  are deemed valid and are utilized in the final output, denoted as  $Q_a \leftarrow Q_{a,a}$ . This selective process ensures that the model focuses on the most relevant Q-values.

In practice, we employ stochastic gradient descent to minimize the loss functions (29) and (30) while training the network. Furthermore, we utilize soft replacement to update the parameters of the target network, which effectively guides the optimization process and ensures the stability of network updates.

Specifically, we employ the CTDE framework, which consists of two classes of agents engaged in collaborative learning. In the **training phase**, illustrated in Fig. 3, agents gather and share observed state information, allowing them to optimize their strategies and learning processes from a global perspective. This centralized training approach ensures that all agents contribute to the optimization of the global objective function using collective information. During this phase, the agents learn cooperative strategies that maximize the global reward  $G_t$ . In the **execution phase**, depicted in Fig. 4, the trained network is deployed on the individual agents, which make decisions independently based on their observed states and learned policies without relying on information from other agents. This significantly reduces the communication and computational burden during online execution. The specific

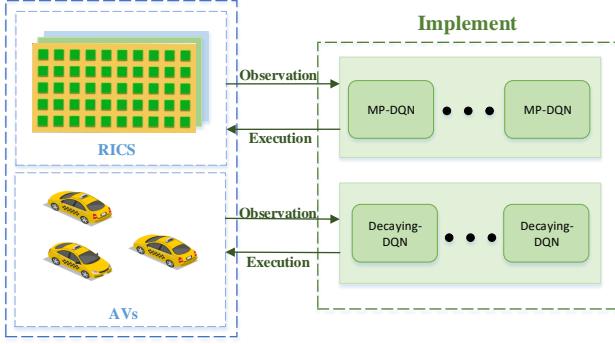


Fig. 4: The implementation phase of the DS-MADRL algorithm.

training process is summarized in **Algorithm 1**.

## V. NUMERICAL RESULTS AND DISCUSSION

In this section, we validate the effectiveness of the proposed DS-MADRL for the autonomous driving network optimization problem. We assume each agent has perfect Channel State Information (CSI), which is updated in every time slot.

### A. Simulation Setup

In the considered simulation scenario, as shown in Fig. 5, one BS is located at  $(0, 0)$  as the central. RICSs are uniformly distributed in a circular region with a radius of 80 meters. The initial positions of the AVs and V2V pairs are randomly distributed within a rectangular area of 100 meters in length and 40 meters in width, at distances ranging from 250 to 350 meters from the origin. The vehicles move at a speed of  $10m/s$ . Specifically, each cell is equipped with one RICS,  $U$  AVs, and  $V$  V2V pairs, with parameters related to the channel and noise detailed in Table II. The network-related parameters for our proposed algorithm are presented in Table III.

TABLE II: System model Parameters

Parameter	Value	Parameter	Value
$K$	30	$U$	10
$V$	2	$Q$	2
$h$	2	$\zeta_{u,c}$	[5, 8] GHz
$f_{u,c}$	[1, 5] GHz	$Penalty$	10
$s_{u,c}$	[1, 3] Mbits	$P_{outage}$	0.01
$P_u$	29 dBm	$P_v$	22 dBm
$\gamma_{th}$	2 bps/Hz	$\alpha$	2.5
$F_{u,c}$	50 GHz	$W\xi_0$	-110 dBm
$\lambda$	0.7	$A_B(Q)$	0.8

TABLE III: Network Parameters

Parameter	Value	Parameter	Value
$Episodes$	600	$Step$	200
$Batch\ size$	32	$Learning\ rate$	0.0001
$decay\ rate$	0.999	$memory\ size$	5000
$\varepsilon$	0.999	$\gamma$	0.95

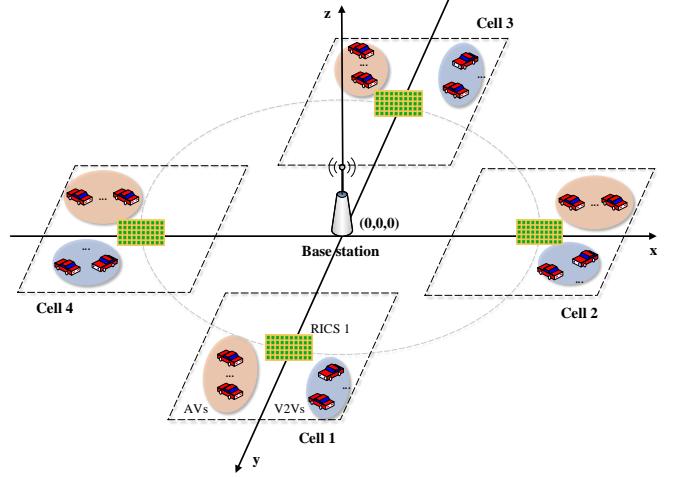


Fig. 5: Simulation setup for RICS-Assisted autonomous vehicular network.

### B. Convergence Performance

This section primarily focuses on the impact of hyperparameters on the DRL algorithm. We employ two comparative algorithms to validate the convergence performance of our proposed algorithm:

- **DDPG+DQN:** We utilize the DDPG algorithm to explore the optimal policy for the continuous variable offloading ratio, which is managed using the MP-DQN in our algorithm.
- **MP-DQN+DQN:** The choose of  $\Theta$  for RICS is performed using the DQN algorithm.

In the experiments, we employed a three-layer Deep Neural Network (DNN), where the two intermediate layers consist of 64 and 32 hidden neurons, respectively. The input layer is designed to receive state features, while the output layer generates Q-values for the action space. Additionally, we utilized the Rectified Linear Unit (ReLU) activation function to introduce non-linearity into the model, and the Adam optimizer was employed to minimize the loss function effectively.

Under the condition of one cell, Fig. 6a illustrates the convergence of the DS-MADRL algorithm compared to two benchmark algorithms when  $U = 10$  and  $V = 2$ . It is evident that all three algorithms achieve convergence within 200 iterations, with a final total safety factor of [9.03, 8.21, 7.59] and an average safety factor of [90.3%, 82.1%, 75.9%]. Clearly, our algorithm demonstrates superior performance, achieving an improvement of [8.2%, 14.4%] over the two benchmark algorithms. This advantage may be attributed to the lack of a decay strategy in **MP-DQN+DQN**, which results in lower exploration efficiency in complex scenarios compared to the DDQN. Furthermore, the combined optimization complexity of the **DDPG+DQN** algorithm is relatively high, as this fully continuous optimization method relies on gradient updates of the policy and can struggle with training difficulties and performance degradation due to the large action space. In contrast, the MP-DQN conducts a forward pass for each discrete action, allowing for more precise optimization of

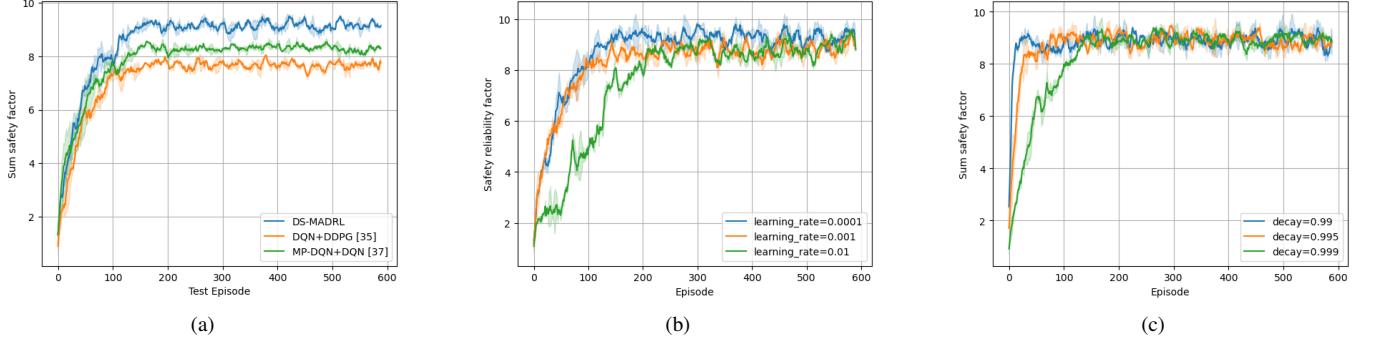


Fig. 6: The convergence of different algorithms is shown in (a), the influence of different learning rates for convergence is shown in (b), and the effect of different decay rates is shown in (c).

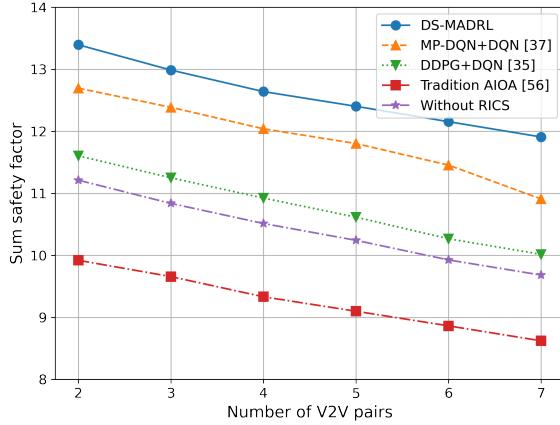


Fig. 7: Sum safety factor of AVs versus varying V2V pairs.

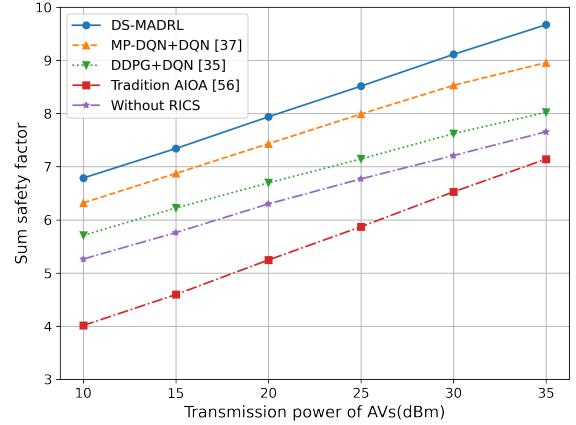


Fig. 8: Sum safety factor of AVs versus varying transmission power  $P_u$ .

action parameters within high-dimensional action spaces.

Fig. 6b compares the impact of varying learning rates on algorithm convergence under identical experimental conditions. It is observed that a lower learning rate of 0.0001 facilitates smoother model updates. A learning rate of 0.001 also achieves stable convergence, although its convergence performance is inferior to that of 0.0001. In contrast, a learning rate of 0.01 results in slower and less stable convergence.

Fig. 6c explores the impact of different decay rates on algorithm convergence. This parameter governs the trade-off between “exploration” and “exploitation” when the agent selects actions. “Exploration” indicates the agent randomly chooses actions to discover potentially better strategies, while “exploitation” signifies that the agent chooses the currently known optimal action. A decay rate that is too low may result in insufficient exploration of new possibilities, making it susceptible to local optima. Additionally, the safety factors correspond to the three decay rates, which are about [89.5%, 89.7%, 90.4%]. A lower decay rate allows for a quicker shift to the exploitation phase during decision-making, relying on the current best strategy. However, the convergence results are not as favorable as those achieved with a larger decay rate. Conversely, a larger decay rate means the agent

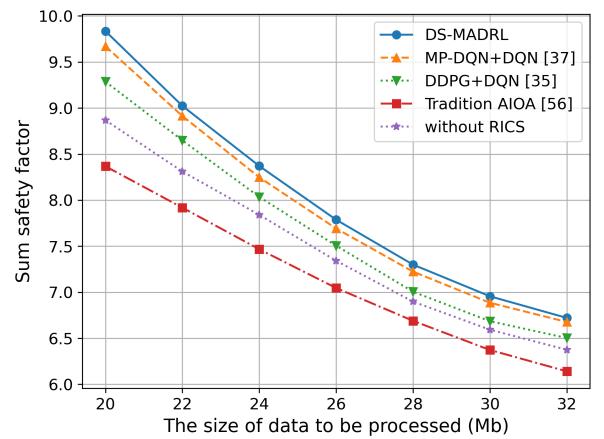


Fig. 9: Sum safety factor of AVs versus varying computation data size  $s_u$ .

must spend more time exploring during the initial phase, leading to a slower convergence speed.

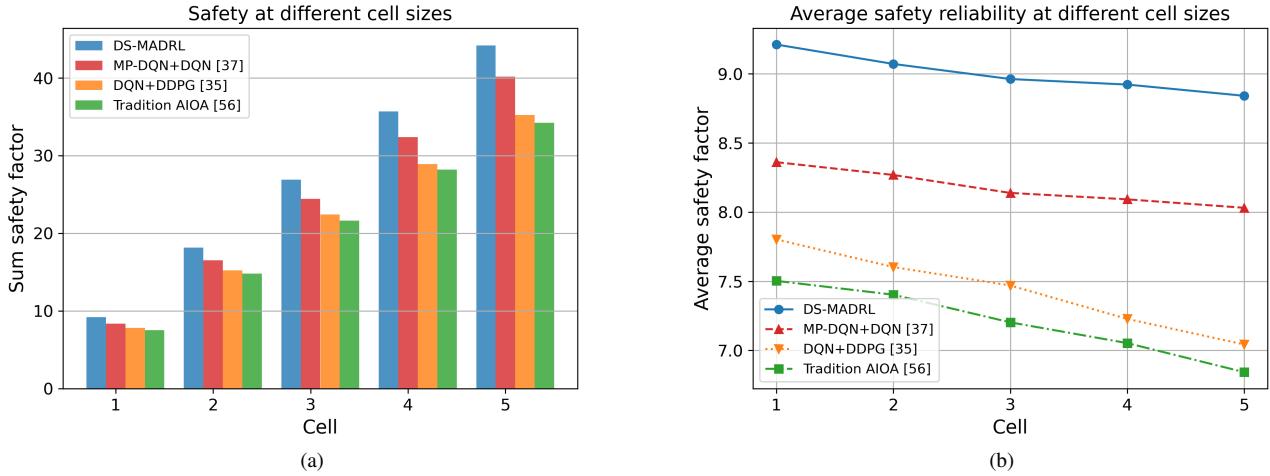


Fig. 10: Sum safety factor versus the number of cells is shown in (a), the average safety factor is shown in (b) in our system.

### C. Results for Single-Cell Systems

The experimental results indicate that the choice of hyperparameters significantly impacts the convergence speed and final performance of the proposed algorithm. Therefore, selecting an appropriate combination of hyperparameters is crucial to ensure that the algorithm converges quickly and reaches a high-performance ceiling, thereby effectively enhancing safety.

In this subsection, we explore the impact of various factors—including the transmission power of AVs, the number of V2V pairs, and the size of data to be processed by AVs in one cell on the system performance are explored separately.

- **Traditional AIOA:** The traditional convex optimization algorithm alternating optimization method is used, whose core idea is to decompose the original problem into three subproblems and optimize only one of them in each iteration until the whole algorithm converges.
- **Without RICS:** Our network scenario does not include assistance from RICS. The algorithm used is still DS-MADRL.

Fig. 7 illustrates a slight decrease in the safety factor of AVs ( $U=15$ ) as the number of V2Vs increases. This phenomenon can be attributed to the multiplexing of spectral resources between the V2V communication link and the V2I link, which introduces significant interference, leading to a deterioration in link quality and a subsequent reduction in throughput. Similar to the conclusion in the previous subsection, the proposed algorithm continues to outperform the other three DRL algorithms and demonstrates a substantial advantage over conventional optimization algorithms, achieving a 17% improvement in the average safety factor of the AVs.

Meanwhile, as depicted in Fig. 8, an increase in the transmission power of AVs directly enhances the signal strength of the V2I links, thereby mitigating the effects of path loss on link quality. A higher V2I signal power also improves the SINR, resulting in enhanced reliability of the link, and the data rate and computational efficiency. Additionally, the conventional

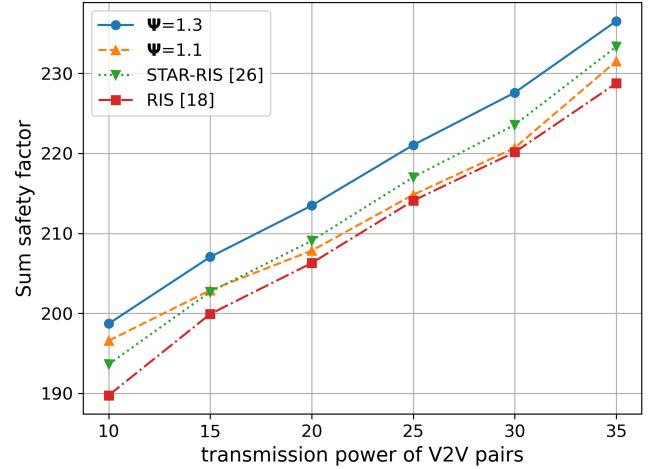


Fig. 11: Performance of different RICS  $\Psi$  under varying  $P_v$ .

optimization algorithms perform significantly worse than the DRL algorithm.

Fig. 9 examines the safety performance of AVs in different data sizes and algorithms. As the  $s_u$  increases, there is an overall decreasing trend in the safety factor, with a more rapid decline observed when the data size is smaller. The rate of decrease tends to stabilize with a gradual convergence trend as the data size continues to increase. Furthermore, the performance of the DS-MADRL algorithm consistently outperforms that of the comparison algorithms. Initially, as the data size increases, the demand for data transmission occupies the bandwidth of the V2I link, leading to the introduction of V2V interference. This results in a sharp decline in channel quality and a corresponding decrease in the safety factor. As the task volume increases further, the critical resources in the system approach the upper limit and the channel quality and interference reach a relatively balanced state.

#### D. Results for Multi-Cell Systems

Fig. 10 characterizes the total safety factor of AVs across varying cell configurations. The left panel illustrates the cumulative safety factor for all cells, while the right panel depicts the average safety factor per cell. The experimental results indicate that as the number of cells increases, the total safety factor of different algorithms rises significantly. However, the average safety factor per cell experiences a slight decline due to the increased load on the BS and the longer waiting times associated with servicing a greater number of cells. Notably, this decrease is constrained to a range of  $1\% \sim 2\%$ , reflecting the robustness and stability of the algorithm even under high-load conditions. The findings demonstrate that the proposed DS-MADRL algorithm can effectively address the computational resource allocation challenges in multi-cell scenarios, ensuring the security and reliability of the system are maintained.

#### E. Impact of amplitude adjustment factor $\Psi$ for RICS

In this part, we compare the effects of different types of RIS in V2V data rates and verify the role of different amplitude adjustment factors  $\Psi$  on interference mitigation. We compare different  $\Psi$ , as well as STAR-RIS [26] and RIS. The specific benchmarks are as follows:

- **RICS with different values:**  $\Psi = 1.1$ ,  $\Psi = 1.3$ , and  $\Psi = 0.8$ . (all elements of RICS are equipped with the same  $\Psi$ ).
- **STAR-RIS:** The rest of the configuration is the same as RICS, without the signal amplitude adjustment function.
- **RIS:** Only possesses signal reflection capabilities.

Fig. 11 illustrates the impact of different amplitude adjustment factors on the V2V data rate. As expected, it is observed that the data rate of V2V pairs increases with the rising of  $P_v$ . Additionally, different amplitude adjustment factors  $\Psi$  have varying effects on V2V data rates; appropriate configurations can effectively mitigate interference from the V2I link. For instance,  $\Psi = 1.3$  is particularly effective in mitigating such interference. While certain parameter configurations may negatively impact interference mitigation. Notably, when  $\Psi = 1.1$ , the V2V pair data rate is inferior to that of the other configurations.

Moreover, well-chosen  $\Psi$  values outperform STAR-RIS and traditional RIS, which lack interference cancellation capabilities. Among the tested configurations,  $\Psi = 1.3$  achieves the best performance, yielding improvements in data transmission rates of [1.97%, 2.36%, 3.43%] compared to the other four schemes. This demonstrates that RICS can leverage its signal adjustment capabilities to effectively mitigate interference experienced by V2V communications, thereby enhancing data rates and system security.

## VI. CONCLUSIONS

This paper studied RICS-assisted autonomous driving under safety requirements and presented a novel DS-MADRL scheme. The considered design optimization problem was modeled as an MG process, utilizing MP-DQN to handle the

continuous-discrete hybrid action space of AVs and employing DDQN for the discrete phase configuration selection of RICS. The proposed approach enabled effective joint decision-making through collaborative interactions with the environment. The convergence performance of the presented joint learning framework was investigated via extensive simulation experiments, which also unveiled the impact of various system parameters on the overall performance. It was demonstrated that the proposed approach maintains robustness and adaptability across different cell scenarios, enhancing significantly the overall performance of the system.

## REFERENCES

- [1] Jihong, X. I. E., Z. H. O. U. Xiang, and Lu CHENG, "Edge Computing for Real-Time Decision Making in Autonomous Driving: Review of Challenges, Solutions, and Future Trends," *International Journal of Advanced Computer Science & Applications* 15.7 (2024).
- [2] F. A. Butt, J. N. Chattha, J. Ahmad, M. U. Zia, M. Rizwan and I. H. Naqvi, "On the Integration of Enabling Wireless Technologies and Sensor Fusion for Next-Generation Connected and Autonomous Vehicles," *IEEE Access*, vol. 10, pp. 14643-14668, 2022.
- [3] S. Han, F. -Y. Wang, G. Luo, L. Li and F. Qu, "Parallel surfaces: Service-Oriented V2X communications for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 11, pp. 4536-4545, Nov. 2023.
- [4] Y. Asabe, E. Javanmardi, J. Nakazato, M. Tsukada and H. Esaki, "AutowareV2X: Reliable V2X communication and collective perception for autonomous driving," *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Florence, Italy, 2023, pp. 1-7.
- [5] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning", *Robot. Auto. Syst.*, vol. 114, pp. 118, Apr. 2019.
- [6] C. Chen, Y. Zeng, H. Li, Y. Liu and S. Wan, "A Multihop Task Offloading Decision Model in MEC-Enabled Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3215-3230, 15 Feb.15, 2023.
- [7] G. Sun, Z. Wang, H. Su, H. Yu, B. Lei and M. Guizani, "Profit Maximization of Independent Task Offloading in MEC-Enabled 5G Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 16449-16461, Nov. 2024.
- [8] A. Mondal, D. Mishra, G. Prasad, and G. C. Alexandropoulos, "Multi-agent reinforcement learning for offloading cellular communications with multiple cooperating UAVs," *arXiv preprint:2402.02957*, 2024.
- [9] T. Zhang, B. Yang, Z. Yu, X. Cao, G. C. Alexandropoulos, Y. Zhang, and C. Yuen, "Anticipatory computation offloading for MEC-enabled vehicular networks via trajectory prediction," *IEEE Int. Conf. Ubiquitous Intel. Comp.*, Denarau Island, Fiji, 2-7 Dec. 2024.
- [10] M. Merluzzi, F. Costanzo, K. D. Katsanos, G. C. Alexandropoulos, and P. Di Lorenzo, "Power minimizing MEC offloading with probabilistic QoS constraints for RIS-empowered communication systems," in *Proc. IEEE Global Commun. Conf.*, Rio de Janeiro, Brazil, Dec. 2022.
- [11] T. Q. Duong, V. N. Q. Bao, G. C. Alexandropoulos, and H.-J. Zepernick, "Cooperative spectrum sharing networks with AF relay and selection diversity," *Electron. Lett.*, vol. 47, no. 20, pp. 1149-1151, Sep. 2011.
- [12] T. Q. Duong, V. N. Q. Bao, H. Tran, G. C. Alexandropoulos, and H.-J. Zepernick, "Effect of primary networks on the performance of spectrum sharing AF relaying," *Electron. Lett.*, vol. 48, no. 1, pp. 25-27, Jan. 2012.
- [13] S. Vassilaras and G. C. Alexandropoulos, "Optimizing access mechanisms for QoS provisioning in hardware constrained dynamic spectrum access," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, Edinburgh, UK, Jul. 2016.
- [14] N. I. Miridakis, T. A. Tsiftsis, G. C. Alexandropoulos, and M. Debbah, "Simultaneous spectrum sensing and data reception for cognitive spatial multiplexing distributed systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3313-3327, May 2017.
- [15] Q. Wu and R. Zhang, "Towards Smart and Reconfigurable Environment: Intelligent Reflecting Surface Aided Wireless Network," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 106-112, January 2020.
- [16] E. Basar, G. C. Alexandropoulos, Y. Liu, Q. Wu, S. Jin, C. Yuen, O. Dobre, and R. Schober, "Reconfigurable intelligent surfaces for 6G: Emerging applications and open challenges," *IEEE Veh. Technol. Mag.*, vol. 19, no. 3, pp. 27-47, September 2024.

- [17] G. C. Alexandropoulos *et al.*, "RIS-enabled smart wireless environments: Deployment scenarios, network architecture, bandwidth and area of influence," *EURASIP J. Wireless Commun. Netw.*, 103, pp. 1–38, Oct. 2023.
- [18] X. Cao *et al.*, "Massive Access of Static and Mobile Users via Reconfigurable Intelligent Surfaces: Protocol Design and Performance Analysis," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 4, pp. 1253–1269, April 2022.
- [19] X. Cao *et al.*, "Reconfigurable Intelligent Surface-Assisted Aerial-Terrestrial Communications via Multi-Task Learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3035–3050, Oct. 2021.
- [20] X. Cao *et al.*, "Reconfigurable Intelligent Surface-Assisted Aerial-Terrestrial Communications via Multi-Task Learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3035–3050, Oct. 2021.
- [21] L. You, J. Xiong, D. W. K. Ng, C. Yuen, W. Wang and X. Gao, "Energy Efficiency and Spectral Efficiency Tradeoff in RIS-Aided Multiuser MIMO Uplink Transmission," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1407–1421, 2021.
- [22] L. You, J. Xu, G. C. Alexandropoulos, J. Wang, W. Wang, and X. Gao, "Energy efficiency maximization of massive MIMO communications with dynamic metasurface antennas," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 393–407, Jan. 2023.
- [23] K. Wang, B. Yang, Z. Yu, X. Cao, M. Debbah and C. Yuen, "Filtering Reconfigurable Intelligent Computational Surface for RF Spectrum Purification," *IEEE Network*, vol. 39, no. 1, pp. 63–70, Jan. 2025.
- [24] J. An *et al.*, "Stacked Intelligent Metasurfaces for Efficient Holographic MIMO Communications in 6G," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2380–2396, Aug. 2023.
- [25] G. C. Alexandropoulos *et al.*, "Hybrid reconfigurable intelligent metasurfaces: Enabling simultaneous tunable reflections and sensing for 6G wireless communications," *IEEE Veh. Technol. Mag.*, vol. 19, no. 1, pp. 75–84, Mar. 2024.
- [26] X. Mu, Y. Liu, L. Guo, J. Lin and R. Schober, "Simultaneously Transmitting and Reflecting (STAR) RIS Aided Wireless Communications," *IEEE Transactions on Wireless Communications*, vol. 21, no. 5, pp. 3083–3098, May 2022.
- [27] X. Gu, W. Duan, G. Zhang, Y. Ji, M. Wen and P. -H. Ho, "Socially Aware V2X Networks With RIS: Joint Resource Optimization," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6732–6737, June 2022.
- [28] X. Gu *et al.*, "Intelligent Surface Aided D2D-V2X System for Low-Latency and High-Reliability Communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 11624–11636, Nov. 2022.
- [29] B. Yang, X. Cao, J. Xu, C. Huang, G. C. Alexandropoulos, L. Dai, M. Debbah, H. V. Poor, and C. Yuen, "Reconfigurable intelligent computational surfaces: When wave propagation control meets computing," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 120–128, Jun. 2023.
- [30] Bertsekas D P. "Constrained optimization and Lagrange multiplier methods". *Academic press*, 2014.
- [31] C. Xing, S. Xie, S. Gong, X. Yang, S. Chen and L. Hanzo, "A KKT Conditions Based Transceiver Optimization Framework for RIS-Aided Multiuser MIMO Networks," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2602–2617, May 2023.
- [32] M. A. ElMossallamy, K. G. Seddik, W. Chen, L. Wang, G. Y. Li and Z. Han, "RIS Optimization on the Complex Circle Manifold for Interference Mitigation in Interference Channels," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6184–6189, June 2021.
- [33] Q. Wu and R. Zhang, "Intelligent Reflecting Surface Enhanced Wireless Network via Joint Active and Passive Beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019.
- [34] R. Zhong, Y. Liu, X. Mu, Y. Chen, X. Wang and L. Hanzo, "Hybrid Reinforcement Learning for STAR-RISs: A Coupled Phase-Shift Model Based Beamformer," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2556–2569, Sept. 2022.
- [35] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, arXiv:1509.02971.
- [36] Xiong J, Wang Q, Yang Z, et al. "Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space". arXiv preprint arXiv:1810.06394, 2018.
- [37] Bester C J, James S D, Konidaris G D. "Multi-pass q-networks for deep reinforcement learning with parameterised action spaces". arxiv preprint arxiv:1905.04388, 2019.
- [38] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," *Neurocomputing*, vol. 190, pp. 82–94, May 2016.
- [39] Y. Dai, Y. L. Guan, K. K. Leung and Y. Zhang, "Reconfigurable Intelligent Surface for Low-Latency Edge Computing in 6G," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 72–79, December 2021.
- [40] G. C. Alexandropoulos *et al.*, "Pervasive machine learning for smart radio environments enabled by reconfigurable intelligent surfaces," *Proc. IEEE*, vol. 110, no. 9, pp. 1494–1525, Sep. 2022.
- [41] P. S. Aung, Y. M. Park, Y. K. Tun, Z. Han and C. S. Hong, "Energy-Efficient Communication Networks via Multiple Aerial Reconfigurable Intelligent Surfaces: DRL and Optimization Approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 3, pp. 4277–4292, March 2024.
- [42] P. Ji, J. Jia, J. Chen, L. Guo, A. Du, and X. Wang, "Reinforcement learning based joint trajectory design and resource allocation for RIS aided UAV multicast networks," *Comput. Netw.*, vol. 227, May 2023, Art. no. 109697.
- [43] J. Wu *et al.*, "Resource Allocation for Delay-Sensitive Vehicle-to-Multi-Edges (V2Es) Communications in Vehicular Networks: A Multi-Agent Deep Reinforcement Learning Approach," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1873–1886, 1 April-June 2021.
- [44] K. Guo, M. Wu, X. Li, H. Song and N. Kumar, "Deep reinforcement learning and NOMA-Based multi-objective RIS-Assisted IS-UAV-TNs: Trajectory optimization and beamforming design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 10197–10210, Sept. 2023.
- [45] A. Al-Hilo, M. Samir, M. Elhattab, C. Assi and S. Sharafeddine, "Reconfigurable Intelligent Surface Enabled Vehicular Communication: Joint User Scheduling and Passive Beamforming," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2333–2345, March 2022.
- [46] P. S. Aung, L. X. Nguyen, Y. K. Tun, Z. Han and C. S. Hong, "Deep Reinforcement Learning-Based Joint Spectrum Allocation and Configuration Design for STAR-RIS-Assisted V2X Communications," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 11298–11311, 1 April 1, 2024.
- [47] K. Stylianopoulos *et al.*, "Lyapunov-driven deep reinforcement learning for edge inference empowered by reconfigurable intelligent surfaces," *Proc. IEEE ICASSP*, Rhodes, Greece, Jun. 2023.
- [48] Y. Wang, X. Li, X. Yi and S. Jin, "Joint User Scheduling and Precoding for RIS-Aided MU-MISO Systems: A MADRL Approach," *IEEE Transactions on Communications*, doi: 10.1109/TCOMM.2024.3496745.
- [49] L. Busoniu *et al.*, "A Comprehensive Survey of Multiagent Reinforcement Learning," *IEEE Trans. Sys., Man, Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
- [50] Z. Chu, Z. Zhu, X. Li, F. Zhou, L. Zhen and N. Al-Dhahir, "Resource Allocation for IRS-Assisted Wireless-Powered FDMA IoT Networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8774–8785, 1 June 1, 2022.
- [51] X. Zhang *et al.*, "Reconfigurable Intelligent Computational Surfaces for MEC-Assisted Autonomous Driving Networks: Design Optimization and Analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 1286–1303, Jan. 2025.
- [52] S. Li, B. Duo, X. Yuan, Y. -C. Liang and M. Di Renzo, "Reconfigurable Intelligent Surface Assisted UAV Communication: Joint Trajectory Design and Passive Beamforming," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 716–720, May 2020.
- [53] Q. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang and T. Q. S. Quek, "Design and Analysis of MEC- and Proactive Caching-Based 360° Mobile VR Video Streaming," *IEEE Transactions on Multimedia*, vol. 24, pp. 1529–1544, 2022, doi: 10.1109/TMM.2021.3067205.
- [54] A. Liu, V. K. N. Lau and B. Kananian, "Stochastic Successive Convex Approximation for Non-Convex Constrained Stochastic Optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4189–4203, 15 Aug. 15, 2019.
- [55] Z. Zhu and H. Zhao, "A Survey of Deep RL and IL for Autonomous Driving Policy Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14043–14065, Sept. 2022.
- [56] Y. Chen, Y. Wang, J. Zhang and M. D. Renzo, "QoS-driven spectrum sharing for reconfigurable intelligent surfaces (RISs) aided vehicular networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5969–5985, Sept. 2021.