Deep Reinforcement Learning Based Ultra Reliable and Low Latency Vehicular OCC

Amirul Islam, Nikolaos Thomos, Senior Member, IEEE, and Leila Musavian

***Abstract*—In this paper, we present a deep reinforcement learning (DRL) framework for vehicular optical camera communication (OCC) systems that ensures ultra-reliable and low-latency communication (uRLLC). We first formulate a throughput maximization problem that aims at optimizing speed of vehicles, channel code rate, and modulation order while respecting the uRLLC requirements. We model reliability by satisfying a target bit error rate and latency as transmission latency. To improve the transmission rate and provide high reliability and low latency, our scheme uses low-density parity-check codes and adaptive modulation. We then solve the optimization problem using the actor-critic-based DRL scheme with Wolpertinger framework. We employ a deep deterministic policy gradient algorithm to operate over continuous action spaces. The evaluation confirms that our proposed DRL-based optimization scheme achieves superior performance compared to radio frequency-based communication systems as well as variants of the proposed scheme. Finally, we verify through simulations that our proposed solution can maximize the communication rate while meeting the uRLLC constraints.**

***Index Terms*—DRL, vehicular OCC, uRLLC, LDPC codes, actor-critic, DDPG**

I. Introduction

Autonomous vehicles (AVs) are guiding the evolution of future smart cities and are regarded as the most transformative technology for intelligent transportation systems (ITSs). AV communication can improve the overall driving experience and ensure traffic s afety b y f acilitating n ew s ervices, e.g., autonomous driving and collision avoidance [1]. In order to realize this, we need to design systems that guarantee the availability of ultra-reliable communication links at extreme low latency [2]. The need to meet both latency and reliability requirements simultaneously makes vehicular communication a very challenging problem as they cannot be ensured by employing current vehicular communication systems. This happens because the current methods attempt to achieve ultra-reliable and low-latency communication (uRLLC) by either using cellular or radio frequency (RF)-based communication systems. These methods typically incorporate edge or remote servers and centralized base stations, relying on centralized resource management. However, the computational resources of these systems are limited; hence, they can be overloaded when there are frequent requests for AV tasks. Besides, RF communication channels are exposed to interference and noise. These render the use of RF- or cellular-based systems problematic for realizing uRLLC in ITS.

On the contrary, in recent years, visible light communications (VLCs) is anticipated to be an essential alternative to RF systems for next-generation communication services [3]. VLC employs light-emitting diodes (LEDs) as transmitters, and photodiodes (PDs) or cameras as receivers. VLC systems employing PDs as receivers are named light-fidelity (LiFi), whereas those using cameras are called optical camera communication (OCC) systems. PDs have very quick responses to signal reception though being non-imaging small devices. However, there is a trade-off in performance between signal reception and transmission signal coverage. Cameras can mitigate the challenges faced by PD-based systems [4]. The rapid advancements in OCC have rendered this system a favourable technology for AV communication [4], [5]. However, meeting the uRLLC constraints necessitate the use of channel coding. Low-density parity-check (LDPC) code is a promising candidate for uRLLC, which has been adopted in the fifth-generation (5G) new radio (NR) services [6]. We use LDPC codes in our system because they can help achieve a high transmission rate, low latency, and high reliability.

Since vehicular environments are time-varying and dynamic, it is challenging to respect uRLLC constraints. Further, vehicular communication systems become even more complex when they involve the control of various decision-making parameters, e.g., channel code rates, speed, distances, and modulation schemes. It is difficult to solve these problems using traditional methods, such as dynamic programming or exhaustive search because of their inherent complexity and time required to solve them. Deep reinforcement learning (DRL) has emerged as a possible candidate to solve autonomous vehicular problems [7], [8]. DRL overcomes partially the complexity of these systems as it can be applied distributively [9]. DRL uses a deep Q network (DQN) which aims at approximating the Q-value of Q-learning algorithm. Besides, DQN cannot be easily applied to continuous problems [10], which is the case for our proposed vehicular system. This is because the DQN maximizes the action-value function to find the appropriate action, whereas it is an iterative optimization process applied at every step in continuous problems. One of the approaches to solve continuous problems is to discretize the state and action spaces. However, this introduces suboptimality, as we may not find the optimal action because of discretization. This happens as inexperienced discretization needlessly discards information, which can be critical for solving the underlying problems.

Amirul Islam is with the Department of Electrical and Electronics Engineering, American International University-Bangladesh (AIUB), Dhaka 1229, Bangladesh (e-mail: amirul@aiub.edu).

Nikolaos Thomos and Leila Musavian are with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: nthomos@essex.ac.uk; leila.musavian@essex.ac.uk).

This research was funded in part by the Engineering and Physical Science Research Council, UK [grant number: EP/X012204/1].

The above issues can be alleviated by adopting the actor-critic based DRL frameworks [10], where the DRL agent incorporates two networks, namely, the actor network and the critic network. The actor network controls the agent's behaviour by selecting actions, whereas the critic network refines the actor's choices to determine the optimal policy approximation. The Wolpertinger architecture [11] along with actor-critic network converges faster than the vanilla actor-critic method over a large actions space by considering the nearest neighbour's actions of a proto-actor action selected by the actor network.

In this paper, we present an actor-critic based DRL approach in vehicular OCC that aims at maximizing the throughput while respecting the uRLLC constraints. In doing this, we optimize the throughput by selecting the optimal code rate, modulation scheme and speed of the vehicle. Our scheme respects uRLLC constraints by designing the reward so that the violation of the constraints results in zero rewards, incentivizing vehicles to avoid them. We train the proposed model using deep deterministic policy gradient (DDPG) [10]. We summarize the key novel contributions of this paper as follows:

- In this study, we use 5G NR LDPC codes in vehicular OCC to ensure uRLLC, which is, to the best of our knowledge, the first work in this regard.
- We present a DRL-based throughput maximization scheme by simultaneously controlling LDPC code rate selection, modulation level, and vehicle speed. This novel joint optimization approach, not previously explored in the context of OCC systems, enhances communication link performance.
- We define our vehicular OCC problem as a Markov decision process (MDP) by uniquely designing the state, action, and reward. The reward is modelled such that it respects uRLLC constraints while maximizing the throughput. The complexity involved in the studied problem by simultaneously optimizing critical parameters (channel coding rate, modulation selection and vehicle speed) in a dynamic vehicular environment makes it particularly challenging. Traditional research might address these aspects individually. Our paper tackles them together, aiming for a more complete approach.
- We propose a novel application of an actor-critic RL framework with a Wolpertinger architecture for OCC, which is particularly suitable for the considered problem due to the use of the K-nearest neighbours (KNN) component. It can handle large action spaces — a challenge often faced in RL due to the large number of state-action variables. This machine learning framework, well-suited to complex decision problems like those in vehicular communication, has not been previously used in Vehicular OCC and remains understudied.
- We model our vehicular system in the Simulation of Urban Mobility (SUMO) simulator, a popular platform for transportation modelling, where the vehicular environment is modelled similarly to the real-world environment. This realistic simulation is a critical aspect of our approach and this paper is the first to use SUMO in Vehicular OCC.
- We validate our proposed methods through extensive simulations by comparing them with RF systems as well as variants of our scheme with regard to achievable throughput, bit error rate (BER), and transmission latency. The results show that our proposed actor-critic based DRL scheme achieves promising performance, maximizes the transmission rate while satisfying the uRLLC constraints, and outperforms the schemes under comparison, showcasing the practical benefits and real-world applicability of our proposed approach.

The remainder of this paper is organized as follows. Section II presents the related works. We then outline the OCC system model and the performance-defining parameters in Section III, while we formulate the maximization problem and RL framework in Section IV. Section V introduces the actor-critic DRL framework using Wolpertinger policy architecture. The simulation setup is provided in Section VI followed by Section VII, where we provide the simulation results for different performance parameters and comparison with various schemes under consideration. Finally, we have drawn conclusions in Section VIII.

II. Related Works

Achieving reliability and latency requirements concurrently complicates AV communications. Several techniques have been proposed in the literature of ITSs to facilitate uRLLC, including delay minimization [12] and reliability guarantee [13]. Specifically, in [12], the authors aim at minimizing the transmission power of vehicular networks by grouping them into clusters, where the violation probability of queuing delay is used to model reliability. In [13], the communication rate is maximized through a joint power control and resource allocation algorithm while considering reliability and latency constraints. Moreover, to reduce latency, edge computing has become an attractive solution, where the requested tasks are processed locally without depending on remote servers [14]. In [14], the authors minimize the computational latency by developing an edge computing strategy in vehicular networks.

The above-mentioned methods lead to interference as they employ RF technology. Various strategies, e.g., machine learning-based schemes [15], frequency planning methods [16], are introduced to mitigate the interference in RF systems. These methods are computationally demanding, where achieving optimality is challenging. More importantly, they have not considered uRLLC requirements and focused only on dealing with interference. OCC has the ability to separate different transmitter sources spatially and process them independently on the image plane. Thus, the receiver can easily discard interference and other light sources, e.g., streetlights, Sun, while focusing on specific pixels where the LEDs strike [4]. There are works in OCC mainly targeting to increase data rate, but they do not consider the uRLLC aspects that we study here. Specifically, in [5], the authors achieved 10 megabits per second (Mbps) data rate by varying LEDs intensity and creating flag images using the communication image pixels in which the high-intensity light sources appear. In [4], the authors achieved a rate of 20 Mbps per pixel without detection of LEDs and a rate of 15 Mbps per pixel with real-time

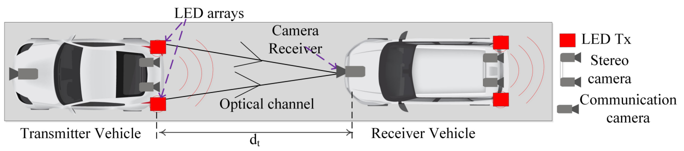


Fig. 1: Proposed vehicular optical camera communication system model.

detection of LEDs in OCC systems. In [17], the rate was further improved to 54 Mbps for BER $< 10^{-5}$ at 50 m distance.

Stochastic optimization schemes cannot be easily applied in an online way in vehicular networks as the networks are dynamic, time-varying and the size of vehicular problems is also bigger, similar to what we consider in this paper. More challenges arise because the problems cannot be solved fast, and, hence, the low latency constraint may be violated, which makes it complicated to respect uRLLC requirements. Deep reinforcement learning has emerged as an optimization framework to deal with the time-varying nature of the optimization problems in vehicular networks [7], [8]. In [18], the authors proposed a solution for joint dynamic channel access and power control using DRL applied in a centralized way. The main drawback of this approach is that centralized training is time-consuming and possibly results in violation of the latency requirements. Moreover, these methods only optimize the spectral efficiency in RF-based systems without considering uRLLC constraints. Since the vehicular environment is time-varying and decision-making parameters, e.g., speed, distance, are continuous, general DQN cannot be trivially applied to this system without discretization of the spaces, which degrades the performance [10]. Recently, actor-critic based DRL frameworks have been studied to solve continuous problems [10]. The Wolpertinger architecture is used in conjunction with the actor-critic framework to limit the search for optimal actions to the nearest neighbour of proto-actor action determined by the actor network [11].

To satisfy the uRLLC requirements in vehicular OCC, channel coding is required in addition to the interference mitigation and DRL framework. Recently, 5G NR LDPC codes have been used to provide reliability and low latency while improving transmission rate [6]. To the best of our knowledge, LDPC code rate optimization using DRL has not been applied in vehicular OCC yet. LDPC codes have already been applied to improve the reliability of communication systems that use adaptive modulation schemes both in wireless [19] and optical communication [20]. These systems use traditional optimization methods to solve the underlying optimization problem, which is inefficient in a time-varying vehicular environment because of the entailed computational complexity. Therefore, in this paper, in order to cope with the diverse nature of the vehicular OCC systems, we follow an actor-critic based DRL framework with Wolpertinger policy architecture to adjust LDPC code rate, speed of vehicles, and modulation scheme while respecting uRLLC constraints.

III. System Modelling

We start this section by introducing the considered vehicular OCC system parameters. We, then, discuss the employed LDPC channel codes and adaptive modulation schemes. We conclude this section by defining different performance parameters of the proposed vehicular OCC system including transmission rates and observed latency.

A. System Overview

Fig. 1 illustrates the proposed model of the vehicular OCC network, where each vehicle is an individual agent. In this system, we define the information-carrying vehicle as "Transmitter Vehicle (TV)" and the information receiving vehicle as "Receiver Vehicle (RV)" while following the TV. The TV transmits information employing rear LED lights, such as, speed, next possible actions (move right, left, stop, or accelerate), position, and safety or action-related information from other vehicles. The RV detects signals from the LED transmitters using a camera. We consider two sets of cameras for each vehicle, one in the front, i.e., a high-speed camera having a frame rate of 1000 frames per second (fps) and another in the back, i.e., a vision camera. The front camera has dual functionality. It first measures the forward distance between the TV and the RV. Then, it receives the transmitted data from the transmitters, i.e., LED lights of TV. The back camera calculates the backward distance, d_t, following a stereo-vision technology as the one discussed in [21]. We would like to note that we have not used two high frame rate cameras both at the frontal and back part of the vehicle because we have only one-way communication with the back LED lights and front camera. Additionally, our setting is cost-efficient because stereo cameras are cheaper than high-speed cameras.

Considering the advantages of adaptive modulation to improve the transmission rates and maintain higher quality of service, we employ M-ary quadrature amplitude modulation (M-QAM) in this paper. However, different modulation schemes can still be applied to our system. M-QAM has already been used in optical communications [22], which offers very low BER, high-speed and flicker-free communication [23]. To further improve the transmission rate and guarantee low BER, i.e., ultra-reliability and low latency, we utilize the 5G NR LDPC code with the M-QAM scheme. We illustrate the overall block diagram of the OCC system employing the transmitter, OCC channel, and receiver in Fig. 2. The transmitter includes an LDPC encoder, an M-QAM modulator, and an LED transmitter, whereas the receiver consists of an image sensor receiver, an M-QAM demodulator, and an LDPC decoder. We will describe the employed LDPC codes in Section III-B. At the transmitter, the data bitstreams are first encoded using LDPC codes before mapping the channel encoded codewords into M-QAM symbols. Then, the coded data are transmitted over an OCC channel through LEDs. At the receiver, the camera captures the modulated light intensity as three different LED states, i.e., on, off, and mid. The originally transmitted information is then extracted from the detected intensity using M-QAM demodulation [24]. For more

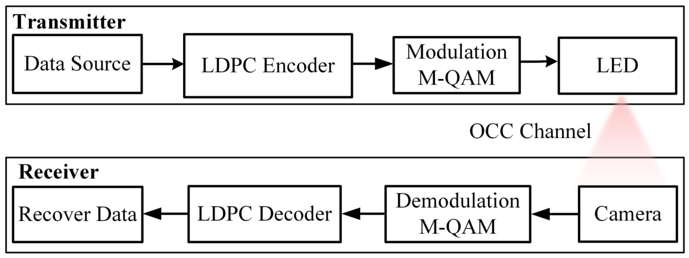


Fig. 2: Block diagram of the employed LDPC coded M-QAM for vehicular OCC.

detail about M-QAM modulation in OCC, please refer to our previous paper [25], where we explain M-QAM encoding and decoding.

As the modulation may change with time, the transmitter should inform the receiver regarding the employed modulation. This is done by appending a small overhead, e.g., some extra bits, in each transmitted packet. In practice, a small set of modulation schemes is used, i.e., 5 in our system. This requires only 3 bits to be appended to the transmitted data for the receiver. Thus, the overhead will be insignificant compared to the transmitted packet size, i.e., 5 kbits, and, thus, we neglect it in our system.

B. Channel Coding

Channel coding strongly impacts the achievable reliability and throughput of a communication system. Since our vehicular OCC system requires ultra-reliability and low-latency, 5G NR LDPC codes that we have used, have already been applied in optical communications [20]. 5G NR systems employ Quasi-cyclic (QC)-LDPC codes as channel coding schemes because of the advantages of efficient implementation and improved performance [26]. The QC-LDPC coded modulation can also resolve the weaknesses of having low reliability and high latency performance for arbitrary order of modulation formats [20], [27] while guaranteeing a low error rate for all code rates. Most notably, 5G NR LDPC codes support a wide selection of data block lengths covering 40 to 8448 bits and diverse code rates, $\varkappa$, between 1/5 and 8/9 [6], [28]. 5G NR codes use a feedback channel to adapt protection, which makes them reliable and efficient. Therefore, we use 5G NR QC-LDPC channel coding over the Galois Field (GF(Q)) for Q-ary QAM transmissions in our vehicular OCC system.

For a GF size of $Q = 2^M$, the transmitter encodes the original data employing Q-ary LDPC codes. Then, the encoded data are transmitted by sequentially mapping them to the M-QAM symbols. On the other hand, the receiver accumulates the modulated symbols, i.e., codeblock, for demodulating and decoding the originally transmitted information. For LDPC decoding, we use the Min-Sum algorithm (MSA) [29]. MSA reduces decoding complexity in LDPC codes by decreasing the number of multiplication operations with only minor performance loss [30]. The receiver uses a standard M-QAM demodulator to demodulate the incoming message in order to recover the original information message.

C. Channel Modelling

In our system, we consider the vehicles to be free from obstruction, and they can communicate with each other continuously through an uninterrupted line-of-sight (LoS) link between the LED transmitter and the camera receiver. OCC has either a diffuse or flat-fading channel. There are two different light propagation in the OCC channel: (i) LoS propagation component that results from the direct light transmitted from transmitter to receiver, and (ii) diffuse propagation component, which is the reflecting lights coming from reflective surfaces or other vehicles. However, the LoS propagation component has higher energy compared to the diffuse component, and hence, in this paper, we neglect the diffuse light component.

The LED has wider directivity and follows a Lambertian radiation pattern. Hence, we can model the emitted light from the transmitters as a generalized Lambertian radiant intensity [25]. Thus, the DC channel gain for visible light LoS link is derived by [25]

$$H_t = \begin{cases} \frac{(m+1)A}{2\pi d_t^2}\ g\, T_s(\theta)\cos^m(\phi)\ \cos(\theta), & 0 \le \theta \le \theta_l \\ 0, & \theta > \theta_l \end{cases} \quad (1)$$

where m represents the Lambertian radiation pattern order, A is the area of the camera lens's entrance pupil, the inter-vehicular distance between the transmitter and receiver vehicle is represented by d_t, g is the lens's gain, $T_s(\theta)$ denotes the signal transmittance of an optical filter, θ is the angle of incidence (AoI) with regard to the receiver, ϕ is the angle of irradiance with respect to the emitter, and θ_l is the field of view (FoV) of the camera lens. We can estimate d_t using a stereo vision camera following a method similar to the one in [21]. In this paper, we consider a fixed AoI, θ, following the analysis of our previous work presented in [31]. In particular, we fix the AoI to 60^o, which can help our system meet the latency and reliability requirements. We made this design decision as it is challenging to change the AoI continuously in a practical scenario because this would introduce additional delays caused by the need to change the AoI mechanically in the vehicle.

In this paper, we ignore the overhead of recognizing the desired light sources under the mobile environment, which is inspired by [32]. We clarify this in our previous paper [25]. The authors in [32] proposed a vehicular motion model in an image plane and showed that the motion of the vehicle along the horizontal and vertical axes on the image sensor plane varies within a single pixel most of the time, which is very small compared to entire image pixels on the captured image. [1]Following [25], the received channel SNR, $\gamma(d)$, for a single LED-camera communication is given as:

$$\gamma(d) = \begin{cases} \frac{k^2P^2}{q\rho P_n W_{\text{fps}} f^2 l^2 d^2}; & \text{if } d < d_c \\ \frac{k^2P^2}{q\rho P_n W_{\text{fps}} s^2 d^4}, & \text{if } d \ge d_c \end{cases} \quad (2)$$

where $k = \rho\frac{(m+1)A}{2\pi} g T_s(\theta)\cos^m(\phi)\cos(\theta)$, P is the optical transmit power, q denotes the electron charge, ρ is the re-

[1]From (1), we see that the channel gain depends on t, which represents the changes in inter-vehicular distance, i.e., d over time. So, from now we adapt d instead of t. From the context, it is also evident that the working variable is distance.

ceiver's responsitivity, P_n represents the power in background light per unit area, W_{fps} is the sampling rate of the camera. f represents the focal length of the lens, and l denotes the diameter of an LED. When the generated image from the LED falls exactly onto one pixel on the image plane, we refer to this distance as critical distance $d_c = fl/s$, where s represents the pixel edge length.

D. Capacity and Latency Modelling

The channel capacity of the OCC systems relies on the applied modulation scheme and employed channel codes. For the LDPC codes with code rate, $\varkappa$, and the M-QAM modulation schemes, the transmission rate is expressed as shown in [25]

$$C(d,\varkappa) = \frac{W_{\text{fps}}}{3} \cdot \frac{\varkappa \, N_{\text{LEDs}} \, w\varrho}{2 \, \tan\left(\frac{\theta_l}{2}\right) d} \cdot \log_2(M(d)) \tag{3}$$

where N_{LEDs} represents the number of LEDs at the transmitter in each row, w is the image width (in case the rolling axis is along the width of the image sensor), ϱ is the size of the LED light in cm^2, and M is the order of the modulation scheme. Please note that, we consider the operation of a rolling shutter camera in our system. The term $W_{\text{fps}}/3$ refers to the fact that the camera must sample the modulated signal three times to decode the original M-QAM signal [24]. In other words, for reconstructing the amplitude and phase perfectly, we need to sample the modulated symbol in three successive frames. Please note that, from hereon we use the terms rate or throughput interchangeably.

We assume that the end-to-end latency of our OCC system is controlled by transmission latency. We ignore the computational latency since we deal with a small volume of data, i.e., the communicated information by TVs to RVs. Consequently, we experience short computational time. Moreover, modern vehicles can be equipped with powerful processors and GPUs and thus are capable of handling significant computational loads, making transmission latency the dominating latency factor. Hence, the transmission latency, $\tau(d,\varkappa)$, for a packet size of L is defined by [25]

$$\tau(d,\varkappa) = \frac{L}{C(d,\varkappa)}. \tag{4}$$

IV. Problem Statement and RL Formulation

A. Proposed Problem Formulation

Considering the vehicular environment and the ultra-reliable and low-latency communication requirements, we formulate a throughput maximization problem for the considered vehicular OCC system. This problem aims at optimizing the modulation schemes and LDPC code rates and adjusting to the vehicle's optimal relative speed. We constrain the BER and latency to respect uRLLC conditions. Hence, we formulate the maximization problem as follows:

$$\max_{\mathcal{M},\,\mathcal{X},\,v} \quad C(d,\varkappa) \tag{5}$$

$$\text{s.t.} \quad \text{BER}(d,\varkappa) \leq \text{BER}_{\max}, \tag{6}$$

$$\tau(d,\varkappa) \leq \tau_{\max}, \tag{7}$$

$$M(d) \in \mathcal{M}, \tag{8}$$

$$\varkappa \in \mathcal{X}, \tag{9}$$

where $\mathcal{M}$ represents the QAM modulation scheme set, $\mathcal{X}$ is the set of LDPC codes, v is the relative speed of the vehicle, BER$_{\max}$ denotes the target BER, and the maximum allowable latency is represented by $\tau_{\max}$. To ensure uRLLC, the reliability is defined by satisfying target BER as in (6), and the latency requirement is respected as in (7). We select the modulation scheme from $\mathcal{M}$ as given in (8). The code rates are adjusted using the set of available 5G NR codes [28], as defined in the IEEE standard as presented in (9). We adapt the distance by $d_t = d_{t-1} + v_t \cdot \Delta t$, where Δt stands for the time difference between the current and previous states and d_{t-1} denotes the distance at the prior state. Our problem belongs to the class of sequential decision problems. This is because the actions of one vehicle affect the actions in the next time slots. Moreover, the decision of changing the speed affects the distance between the agent vehicle and the neighbouring ones, and hence (5) is a sequential problem. It is important to note that by using its back cameras, each agent (vehicle) can accurately estimate the distance from the vehicles behind it. This information is sufficient to solve our problem, as represented by (3) and (4). With this information, the agent vehicle can compute both throughput and latency, and leverage this information to train the OCC system and enhance its overall performance.

The optimization problem in (5) is mixed-integer programming (MIP) with nonlinear constraints for BER in (6) and delay in (7). As a result, our problem is non-deterministic polynomial-time-hard [33]. MIP problems are known to have high computational complexity [34], and while dynamic programming or exhaustive search techniques can be used to solve them, these methods cannot be used in dynamic systems like the one we study in this paper because they are extremely time-consuming and/or computationally demanding. The decision space in our problem is incredibly large since we control the speed, code rate, and modulation. The use of deep RL allows us to solve the problem with less computational and time complexities. Please note that vehicular communication must satisfy the uRLLC requirements to ensure that the information is received reliably within the required time. In RL, the vehicles (agents) cooperate with the unknown environment to decide the optimal policy, i.e., selecting optimal code rate, speed, and modulation order, while adapting to the environmental changes. Before driving to the solution in the following section, we first formulate the problem in (5) - (9) as a MDP in the next subsection.

B. MDP Modelling

The proposed optimization problem in (5) is formulated as MDP, where each vehicle is an agent, and the other vehicles, apart from the specific vehicle, are considered the environment. The agent explores and interacts with the environment to have a better understanding of it and decides the throughput maximization policies based on the observations of the environmental states. The MDP is defined as a tuple ($\mathcal{S}$, $\mathcal{A}$, p, r, ζ) [9], where we define each element as follows:

$\mathcal{S}$: a set of available states; $\mathcal{A}$: a set of available actions; $p(s_{t+1}, r_t | s_t, a_t)$: the transition probability that describes the probability when the agent transits from the state s_t to the next state s_{t+1} by selecting an action a_t from $\mathcal{A}$; r: the reward; and $\zeta \in [0, 1]$: discount factor that affects the future rewards by discounting the impact of the action gradually. A discount factor $\zeta = 0$ provides a short-sighted goal that maximizes the immediate reward. When ζ is close to 1, the agent focuses more on the future reward, and the scheme becomes farsighted. In practice, a farsighted approach is desirable, as it achieves better returns by focusing on future discounted rewards. We now present the definition of state space, action space, and reward function for the studied RL problem as follows:

*1) **State Definition**:* At the current time t, an agent looks at the state s_t from the vehicular environment. In our system, the state consists of three parameters: the backward distance, d_t, the transmitting modulation scheme, M_t, from the set $\mathcal{M} = \{4, 8, 16, 32, 64\}$, and the code rate, $\varkappa_t$, from the set $\mathcal{X} = \{\text{5G NR codes}\}$ [28]. We summarize the state at time t as $s_t = \{d_t, M_t, \varkappa_t\}$.

*2) **Action Definition**:* At each state s_t, the agent selects an action a_t from the set $\mathcal{A}$ following a policy π. For our considered system, the action space is the combination of selecting a modulation scheme from the set $\mathcal{M}$, code rate from set $\mathcal{X}$, and adjusting the speed, v_t. In summary, the action space is expressed as $a_t = \{\triangle M_t, \triangle \varkappa_t, \triangle v_t\}$, where $\triangle$ represents the change of values of the respective parameters, e.g., $\triangle M_t$ refers to the change in modulation scheme.

*3) **Reward Function**:* Following the action taken at the current state, the agent receives a reward. Note that, an effective design of the reward is imperative for the learning algorithm to obtain the desired goal, which is achieved by experience and a multitude of attempts. Hence, the reward needs to be relevant to the objective function. In our framework, the reward is the weighted sum of the rewards corresponding to inter-vehicular distance (d_t), throughput (5), BER constraint (6), and latency constraint (7). Firstly, we model the reward for the distance changes, r_t^{d}, as follows:

$$r_t^{\text{d}} = \begin{cases} d_t - d_{\text{stop}}, & d_t < d_{\text{stop}} \\ \frac{1}{d_t - d_{\text{stop}}}, & d_t > d_{\text{stop}} \end{cases} \tag{10}$$

where d_{stop} denotes the stopping distance that is the summation of braking distance, i.e., the distance that the vehicle travels after triggering the brakes and reaction distance, i.e., the distance that the vehicle travels between the time required by the driver to react after observing any situation [35]. We, then, model the reward for the reliability, i.e., BER, r_t^r, as:

$$r_t^r = \mathbb{1}_{\text{b}}(\text{BER}_t \leq \text{BER}_{\text{max}}), \tag{11}$$

where $\mathbb{1}_{\text{b}}$ stands for the indicator function for the BER, which returns 1 when the BER condition is satisfied or 0 otherwise. Similarly, the latency is constrained so that it meets the low latency requirement. Accordingly, the reward for latency, r_t^τ, is modelled as follows:

$$r_t^\tau = \mathbb{1}_\tau(\tau_t \leq \tau_{\text{max}}), \tag{12}$$

where, $\mathbb{1}_\tau$ is the indicator function for latency that returns 1 for true condition and 0 otherwise.

Finally, from the above modelling, the overall weighted sum of rewards, r_t, is expressed as

$$r_t = \omega_d \, r_t^{\text{d}} + \omega_b r_t^r + \omega_\tau r_t^\tau + \omega_c \, C(d, \varkappa), \tag{13}$$

where, ω_d, ω_b, ω_τ, and ω_c are positive weights that balance the distance, BER, latency, and communication rate rewards. The weights can be adjusted based on the system requirements. For instance, a higher value for ω_c gives higher priority to selecting actions that maximize the throughput at every step. The reward (13) is designed in a manner so that the violation of reliability and delay constraints (Equations (11) and (12), respectively) result in zero reward. This design guides the learning process towards solutions that consistently meet the uRLLC requirements.

The return in the MDP is the discounted sum of future rewards received by the agent, which is expressed as $G_t = \sum_{j=0}^{\infty} \zeta^j r_{t+j+1}$. The RL framework aims at maximizing the expected return over all timesteps, i.e., $\max \ \mathbb{E}[G_t(s_t, a_t)]$, starting from a given state, s_t, and taking an action, a_t, following a policy, π_t, thereafter. The Q-learning-based action-value function is commonly used in RL algorithms. It can be expressed in a recursive relationship using the Bellman equation:

$$Q^\pi(s_t, a_t) = r(s_t, a_t) + \zeta \mathbb{E}_{a_{t+1} \sim \pi} \left[Q^\pi(s_{t+1}, a_{t+1}) \right], \tag{14}$$

where $\mathbb{E}_{a_{t+1} \sim \pi}$ stands for expectation of future accumulated reward $Q^\pi(s_{t+1}, a_{t+1})$, while taking an action following a policy π at time $t+1$. To determine the Q value, the agent considers both the current and next state over all actions $\mathcal{A}$.

V. Proposed Solution

Q-Learning is a well-known method [9] employed for addressing problems formulated as MDP. The size of the state-action space, however, affects the Q-learning algorithm's rate of convergence. The RL agent can quickly explore each state-action pair in a small state-action space and identify the optimal policy. The Q-table size grows incredibly big for a large state-action space, which results in slow convergence times. This happens because the Q-learning agent may not explore numerous state-action pairs in a reasonable time. Although there are several linear function approximation approaches [36], [37] that can be used to solve RL problems, the capabilities of these approaches are limited to medium-sized problems. Despite using efficient approximation functions, traditional RL algorithms cannot quickly learn the informative features of the environment in high-dimensional and complicated systems. This is due to the RL agent's potential failure to explore many state-action pairs. Another disadvantage of the original Q-learning technique is that it requires a significant amount of storage space for the Q-table and a longer time to converge for indefinitely large state-action spaces. The problem quickly becomes insoluble as the cost of evaluating the Q function increases since the execution complexity rises linearly with the number of state-action spaces.

Tabular Q-learning cannot be employed since it only works with discrete variables; nevertheless, distance and speed are continuous variables that result in a large state-action space. Discretization may be applied, however, it has an impact on the solution's quality, where there is a trade-off between the discretization and the size of the state-action space. Thus, we must compromise on performance because we may be required to generalize the state-action space while discretizing them. Unfortunately, Q-learning cannot be straightforwardly applied to continuous problems. This happens because, when we have significantly large action spaces and function approximators, the optimization becomes practically too slow. This motivates the approach that we follow here. In particular, we employ an actor-critic framework based on the DDPG algorithm [38], where we utilize a policy architecture known as Wolpertinger architecture [11]. This policy architecture removes the massive computational cost of evaluating Q-function on each action taken.

A. Algorithm Overview

We have already mentioned that the proposed policy architecture follows the Wolpertinger architecture [11], which is based on actor-critic [9] framework. The dual-network structure collaboratively enables the model to balance exploration and exploitation while continually improving policies to meet the uRLLC requirements. The Wolpertinger architecture has three main components: actor network, K-nearest neighbours, and critic network and is executed in three main steps. The actor first takes states as the input and produces a proto-actor, $\hat{a}$, at the output. Secondly, the proto-actor is fed as input to the KNN, which computes the L_2 distance between the proto-actor and each valid action, i.e., actions that respect latency and reliability constraints, and keeps a list of the K actions that result in the smallest L_2 distance. In this way, the proto-actor is expanded over the action space, $\mathcal{A}_K$, where K is the number of elements and every element is an action $a \in \mathcal{A}$. Finally, the critic network takes $\mathcal{A}_K$ as the input to rectify the decision of the actor network based on the Q value. In order to update the actor and critic networks, we train the policy using the DDPG algorithm [10]. We use multi-layer neural networks for the actor-critic function approximators. A detailed explanation of the essential elements of the actor-critic framework and Algorithm 1 is presented below.

*1) **The actor network**:* The Wolpertinger architecture operates on actions within a continuous space, R^n and then maps this output to the discrete action set $\mathcal{A}$. We define the network through a function which is characterized as θ^μ [11].

$$\begin{aligned} &\mu(s \mid \theta^\mu) : \mathcal{S} \to R^n \\ &\mu(s \mid \theta^\mu) = \hat{a}. \end{aligned} \tag{15}$$

This function provides a proto-action in R^n for a given state, which may not be a valid action, i.e. it is likely that $\hat{a} \notin \mathcal{A}$. Therefore, we need to map $\hat{a}$ to an element in $\mathcal{A}$. We can do this through function g, which performs as follows:

$$g : R^n \to \mathcal{A} \tag{16}$$

*2) **K-nearest neighbours (KNN)**:* For a large action space, the potential high computational complexity can be reduced by the proto-actor generation. However, transforming to only a single actor from large action space dimensions will direct to choosing imperfect decisions. To resolve this, KNN mapping, g_K, is used to expand the actor's action to a valid actions' subset, $\mathcal{A}_K \subset \mathcal{A}$ that are in close proximity of the proto-actor. We express the returned action set, $\mathcal{A}_K$, from g_K as: $\mathcal{A}_K = g_K(\hat{a}_t)$, where

$$g_K = \arg \overset{K}{\min_{a \in \mathcal{A}}} \mid a - \hat{a} \mid^2 . \tag{17}$$

where $\mid a - \hat{a} \mid^2$ represents the features distance between the proto-actor $\hat{a}$ and the chosen action a. After selecting the proto-actor by the actor network, the agent determines the KNN feature distances by roaming over the action space and accordingly, the action set can be formed. We can find the K nearest neighbours using (17). It is worth noting that this lookup process can be performed in an approximate manner with logarithmic time complexity, as demonstrated in [39].

*3) **The critic network**:* To avoid selecting an invalid action or an action that leads to a low Q-value frequently, the critic network is introduced, which refines the actions chosen by the actor. The deterministic policy in the critic network is characterized as follows:

$$Q\left(s_t, a_t \mid \theta^Q\right) = \mathbb{E}\left[r(s_t, a_t) + \zeta Q\left(s_{t+1}, a_{t+1} \mid \theta^Q\right)\right], \tag{18}$$

where θ^Q is the parameters of the critic network. The critic calculates the Q value while considering the current state, s_t, and the next state, s_{t+1}, as the input. The critic network evaluates all actions in $\mathcal{A}_K$ and chooses the action which gives the maximum Q-value, as follows:

$$a_t = \arg \max_{a_t \in \mathcal{A}_K} Q(s_t, a_t \mid \theta^Q). \tag{19}$$

Update: At each timestep, a minibatch is sampled uniformly from the replay memory to update the actor and critic networks. Since DDPG is an off-policy algorithm, it allows the algorithm to benefit from learning across a set of uncorrelated transitions. Hence, we update the actor policy using DDPG with a minibatch size $N_\mathcal{B}$, which is expressed as

$$\begin{aligned} \nabla_{\theta^\mu} J \approx &\frac{1}{N_\mathcal{B}} \sum_t \nabla_a Q\left(s, a \mid \mu^Q\right) \mid_{s=s_t, a=\mu(s_t)} \\ &\nabla_{\theta_\mu} \mu\left(s \mid \theta^\mu\right) \mid s_t, \end{aligned} \tag{20}$$

and the critic is updated by minimizing the loss:

$$L = \frac{1}{N_\mathcal{B}} \sum_t \left(y_t - Q\left(s_t, a_t \mid \theta^Q\right)\right)^2, \tag{21}$$

where

$$y_t = r_t + \zeta Q'\left(s_{t+1}, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{Q'}\right) \tag{22}$$

represents the target value at each iteration.

Implementing (21) directly with neural networks is tricky in most RL problems. The updated $Q(s, a \mid \theta^\mu)$ network is utilized to calculate the target in (22); however, the update of the Q-value suffers from divergence issues. Instead of directly copying the weights, we present a similar target network used in [40] as the solution. This solution is adjusted for actor-critic while using "soft" target updates. In doing so, we calculate the target values by copying the actor and critic networks, $Q'(s, a \mid \theta^{\mu'})$ and $\mu'(s \mid \theta^{\mu'})$, respectively. We then update the target network's weights by slowly tracking the networks as

$$\theta^{Q'} \leftarrow \beta\theta^{Q} + (1-\beta)\theta^{Q'}, \tag{23}$$

$$\theta^{\mu'} \leftarrow \beta\theta^{\mu} + (1-\beta)\theta^{\mu'}, \tag{24}$$

where $\beta \ll 1$ is the soft target update rate. This indicates that we constrain the target values to change them slowly while improving learning stability.

In contrast to the general Q-learning, where the balance between exploration and exploitation is controlled using a ϵ-greedy method [9], the DDPG algorithm deals with the exploration problem separately from the learning algorithm. Hence, we define the exploration policy μ' by sampling the noise, n_t, from a noise process and adding it to the actor policy

$$\mu'(s_t) = \mu(s_t \mid \theta^{\mu}_t) + n_t, \tag{25}$$

where n_t is chosen to suit the environment. We consider temporally correlated noise to explore well in the environment using a similar process to that introduced in [41].

Although there is no general theoretical guarantee that among the nearest neighbours of a proto-actor is the optimal action, numerous studies have demonstrated the efficacy of the KNN algorithm in similar problems [42], [43]. This has been observed through the conducted experiments in the performance evaluation section that demonstrate the effectiveness of our approach.

B. Complexity Discussion

The Wolpertinger algorithm's training process has a time complexity proportional to the amount of training data and time. We, therefore, only focus on the training process. The time complexity of the running process is characterised by the neural networks' architecture as well as the state-action space dimension. Please note that, one fundamental difference between this and traditional stochastic non-convex methods is the inherent computational speed to link the bias in the search direction, which is defined by the method used in the critic network. The convergence rate is determined by the convergence rate of the critic network. Slow convergence in critic networks is the bottleneck for actor-critic, whereas fast convergence methods shift this problem to policy gradient updates. It is demonstrated experimentally that fast critic convergence results in faster actor-critic convergence, but the stationary point it reaches is worse than the methods that guarantee slower convergence [10].

Furthermore, the Wolpertinger architecture's time complexity varies linearly with the percentage of actions selected, i.e., K, from the total actions set. In practice, however, increasing the value of K above a certain point does not result in improved performance [11]. The authors of [11] demonstrate a significant improvement in the performance with higher K values though it renders other performance improvements. When only 5% or 10% of the total number of actions are used, the method performs similarly to when the entire action set is used. Using the remaining actions would result in relatively minor performance gains while significantly increasing computational time. Therefore, in our case, we use 5% to 20% of the available action set for the KNN algorithm.

Algorithm 1 Actor-Critic Algorithm

```
1: Randomly initialize critic network $Q\left(s, a \mid \theta^{Q}\right)$ and $\mu(s \mid \theta^{\mu})$ with weights $\theta^{Q}$ and $\theta^{\mu}$.
2: Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^{Q}$, $\theta^{\mu'} \leftarrow \theta^{\mu}$
3: Initialize SUMO environment and replay memory according to system requirements.
4: for episode do
5:     Receive the initial observation state $s_t$
6:     for each timestep $t$ do
7:         Receive observation state $s_t$
8:         Actor: Receive proto-action from actor network $\hat{a}_t = \mu(s_t \mid \theta^{\mu})$.
9:         KNN: Retrieve $K$ approximately closest actions $\mathcal{A}_K = g_K(\hat{a}_t)$
10:        Critic: Select action $a_t = \arg\max_{a_t \in \mathcal{A}_K} Q(s_t, a_t \mid \theta^{Q})$ according to the current policy
11:        Execute action $a_t$, and observe reward $r_t$ and observe new state $s_{t+1}$
12:        Store transition $(s_t, a_t, r_t, s_{t+1})$ in replay memory.
13:        Sample a random mini-batch of $N_{\mathcal{B}}$ transitions $(s_t, a_t, r_t, s_{t+1})$ from replay memory
14:        Set target $y_t = r_t + \zeta Q'\left(s_{t+1}, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{Q'}\right)$
15:        Update critic by minimizing the loss using (21)
16:        Update the actor policy using the sampled policy gradient with (20)
17:        Update the target networks with $\beta \ll 1$ using (23) and (24)
18:        Update the state, action and reward
19:        Update rate
20:    end for
21: end for
```

VI. Experimental Setup

In this section, we present the simulation setup of the proposed actor-critic-based DRL framework for our vehicular OCC system. In particular, we start by presenting the microscopic traffic simulation of SUMO [44] environment. We, then, provide the considered parameters for the proposed actor-critic scheme and training workflow.

A. SUMO Framework

In order to implement our vehicular environment, we have chosen the SUMO traffic simulator because it: (i) is an open-source, microscopic, multi-model traffic and extensible simulator; (ii) offers flexibility and scalability to create the required scenario maps; and (iii) supports Traffic Control Interface (TraCI), a Python-based application programming interface to adapt the simulation online. There are already various sets of driver models in SUMO, and it is relatively

TABLE I: SUMO modelling parameters

Parameter	Value
Initial velocity of vehicle	5 miles per hour
Window size of the simulation	180 m
Maximum number of vehicle per window	20
Number of lane	1
Step length	1 m
Lateral movement of vehicle	0.64 m per timestep

easy to include additional models. We transform the proposed vehicular environment into a corresponding SUMO map, where each vehicle is an agent. The vehicles randomly enter in the SUMO environment and then move or leave the map following the SUMO mobility model set by the system. The interaction between the SUMO framework and the DRL agent is managed by the middleware, TraCI. The agent can retrieve various features of the vehicle from the SUMO network, such as the inter-vehicular distance, speed of the vehicle, current position of the agent, and so on.

We adjusted the SUMO environment according to the requirements of our proposed multi-agent vehicular system by changing some settings. For example, we have a window size of the simulation of 180 m. In the SUMO model, we add some aggressively moving vehicles to the environment, which gives our system more diversity. The simulation parameters of the SUMO framework are presented in Table I.

With the training being initiated, the vehicles are loaded in the SUMO with the specified settings for our vehicular environment. During the training, TraCI interacts with the SUMO environment and extracts the required data by observing the environment. At the current state, the agent observes the vehicular environment and allocates an action following policy neural network. Thus, the agent updates its state while moving to the next state in the SUMO environment. Then, the reward is computed and communicated to the agent to optimize in every simulation run. This process recommences until all the simulation steps are finished or a convergence threshold is achieved.

B. OCC System Design

To demonstrate the competence of our proposed OCC system, we consider the communication of 10^{11} bits and a packet size of 5 kbits. For our simulation, we consider the 5G NR LDPC codes set from the IEEE standard [45]. The required stimulation parameters are shown in Table II. We train the system model with the transmission of zero codewords, i.e., all the bits of the codeword are zero, which are sufficient for the training as the channel is symmetric [46]. On the transmitter side, the zero codewords are encoded by the LDPC encoder and, after M-QAM modulation, are transmitted through the LoS OCC channel. On the receiver side, the data are first demodulated by the M-QAM demodulator and then decoded by the LDPC decoder. The error is computed by comparing the received codeword with the zero codeword.

Our system aims at meeting the uRLLC constraints in vehicular OCC, so we set threshold values for delay and reliability accordingly. The requirements for uRLLC depend on the use case; e.g., ultra-reliability in terms of packet error rate can range from 10^{-5} to 10^{-9} [47] and low-latency can range from 1 ms to 10 ms [48]. For vehicular communication, the required reliability is 10^{-5} and the latency is 3 ms - 10 ms for a packet of 300 bytes [48]. Hence, for large packet sizes (5 kbits in our case), a maximum of 10 ms latency will meet the requirements of vehicular communication. Regarding the reliability constraint, we have set the BER at a maximum of 10^{-7}, which is measured by communicating a 5 kbits packet.

C. Actor-critic DRL Framework

1) Training Parameters Settings: In this subsection, we introduce the actor-critic-based DRL network settings and the considered training parameters. The individual actor and critic network has three fully connected layers, containing an input layer, an output layer, and a hidden layer. The input layer has $(d + \mathcal{M} + |\mathcal{X}|)$ nodes since the state space combines the distance, modulation scheme, and code rate, where $N_d = 150, |\mathcal{M}| = 5$, and $|\mathcal{X}| = 20$ with $N_d = 150$ being the number of nodes for distance. Whereas the output layer has $(\triangle M + \triangle \varkappa + \triangle v)$ nodes, as in our proposed system, the action includes the change in modulation scheme, code rate, and velocity where $\triangle M = 5$, $\triangle \varkappa = 20$, and $\triangle v = 60$). We consider 250 neurons at the hidden layer. The rectified linear unit (ReLU) is employed as the activation function [49] for both the actor and critic networks. To ensure that the critic network learns faster than the actor network, we have set the learning rate of the actor network as 0.0001 and the learning rate of the critic network as 0.001. Whereas, we set $\beta = 0.001$ to update the soft target value. It is common that we achieve fast convergence when the learning rate is large, but it can also lead to unsatisfactory convergence performance, e.g., local minima, saddle point, concurrently. Contrarily, a small learning rate results in extensive computation requirements and causes slow convergence. In our evaluations, we utilize TensorFlow [50] to realize the DRL algorithm. We use root mean square propagation (RMSProp) optimizer [51] as the training algorithm to minimize the loss function and update DQN network parameters.

In our implementation, we train the actor-critic based DRL scheme for 10000 timesteps, which is sufficient to ensure convergence. For the noise process in exploration, we use temporally varying correlated noise. We utilize the Ornstein-Uhlenbeck noise process models [41] with a mean value equal to 0.15 and variance equal to 0.2 that produces values around zero. The discount factor, ζ, is set to 0.98 for our proposed scheme. We train the network with minibatch sizes of 64 while having a replay buffer size of 10^{11} to store the transitions in the memory. We also perform normalization to bring the different sub-rewards corresponding to distance, BER, latency, and transmission rate in (13) to a similar scale. This normalization improves the performance and provides training stability for the NN model. Specifically, we normalize the reward function of distance (10) and rate of (5) to keep the scale of (13) between 0 and 1. We specify the simulation parameters in Table II.

2) Training Procedure: The training workflow of the proposed actor-critic based DRL algorithm is summarized in Algorithm 1. In every training step, an agent observes the

TABLE II: Simulation Parameters

Parameter, Notation	Value	Parameter, Notation	Value
Angle of irradiance w.r.t. the emitter, ϕ	70^o	Number of LEDs at each row, N_{LEDs}	30
AoI w.r.t. the receiver axis, θ	60^o	Packet size, L	5 kbits
FoV of the camera lens, θ_l	90^o	Size of the LED, ϱ	15.5 $\times$ 5.5 cm^2
Image sensor physical area, A	10 cm^2	Resolution of image, w	512 $\times$ 512 pixels
Optical filter's transmission efficiency, T_s	1	Mini-batch size, $N_{\mathcal{B}}$	64
Lens gain, g	3	Replay memory size	10^{11}
Optical transmitting power, P	1.2 Watts	Number of hidden layer (Neurons)	1(250)
Electron charge, q	1.6 $\times 10^{-19}$ C	Discount factor, ζ	0.98
Background light power per unit area, P_n	6000	Optimizer	RMSProp
Diameter of a LED, l	6 mm	Activation function	ReLU
Focal length of camera lens, f	21 mm	Learning rate (Actor network, Critic network)	$0.0001, 0.001$
Edge-length of a pixel, s	7.1 μm	Soft target updates rate, β	0.001
Constellation set, $\mathcal{M}$	4, 8, 16, 32, 64	Gradient momentum (used by RMSProp)	0.95
Camera-frame rate, W_{fps}	1000 fps	Interval between two timesteps	1 second
Distance range	0-150 m	Code rate range	5G NR standard [28]

state s_t (distance, modulation scheme, and code rate) on line 5. Then, the actor network finds a proto-actor following the policy on line 8, which is then expanded to an action set $\mathcal{A}_K$ via KNN on line 9. The critic network evaluates the action (change in modulation, code rate, and velocity) and finds the action set that can provide the maximum state value on line 10. This action is then applied to the environment on line 11 and the resulting reward and subsequent state are stored along with the applied action (s_t, a_t, r_t, s_{t+1}) in the replay buffer after each epoch on line 12. On line 13, a random transition is sampled from the replay buffer, and line 14 performs Q-learning update by applying (22), using the target network's weights for the target Q. On line 15, the critic parameter is updated by minimizing the loss (21) whereas the actor is then trained on line 16 by following the policy gradient using (20). Subsequently, the target network is updated by slowly varying the weights of (23) and (24) on line 17. This allows the learning algorithm to leverage otherwise ignored information of which action was actually executed for training the critic while taking the policy gradient at the actual output.

VII. Performance Evaluation

In this section, we conduct numerous simulations to examine the performance of our proposed DRL-based throughput maximization scheme in the vehicular OCC system. We begin this section by outlining the various comparison schemes under consideration. Then, we assess several performance metrics, such as throughput, latency, and BER for all schemes under comparison.

A. Comparison Schemes

We analyze the performance of our proposed actor-critic-based DRL scheme, termed hereafter as the proposed scheme, against different methods to get insights into the system performance. We present a brief summary of all the schemes under comparison below:

- **Proposed scheme**: By the proposed scheme, we refer to our DRL-based vehicular OCC system, where each vehicle is an agent considering other vehicles as the environment. In this case, we employ the settings as we discuss in Section VI. The discount factor ζ is set to 0.98. We demonstrate the effect of the selected number of neighbours by examining different K values for the employed KNN framework.

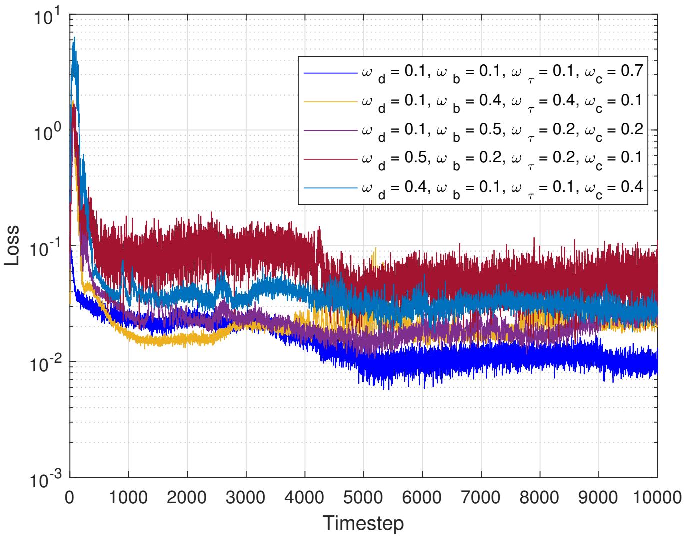


Fig. 3: Convergence of loss function for different weight settings of sub-reward function.

- **Greedy**: This is one of the variants of our scheme, where the discount factor ζ is set to 0 in (21), while other system parameters are kept the same, as noted in Table II. In this scenario, the agent chooses the action which maximizes only the immediate reward.
- **Farsighted**: This method is a variant of our scheme, where we set the discount factor ζ is set to 1 in (21), while other system parameters are kept the same, as noted in Table II. This scheme focuses on future rewards and ignores immediate rewards.
- **RF-based Scheme [7]**: This is a RF technology-based resource allocation scheme presented in [7]. For this scheme, we adapt the hyper-parameters according to our proposed scheme while keeping the environment unchanged. This scheme considers centralized learning, which involves communication between server and agent. This system incurs extra delay due to having feedback loop.

Please note that for the proposed scheme, we consider three different K values, whereas, for greedy and farsighted variants of our scheme, we set $K = 0.1$. This is because it provides a good compromise between performance and convergence speed, as we will see in the sequel. Please note that, throughout our simulation, we refer to timestep

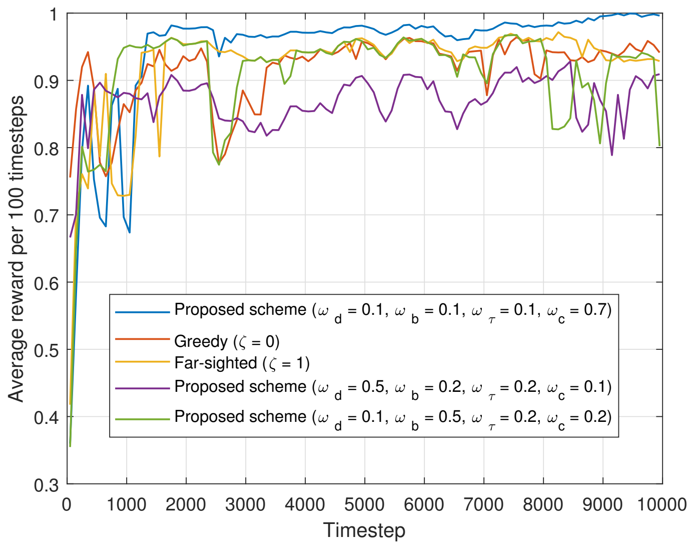


Fig. 4: Average reward per 100 training timesteps for the proposed scheme and its variants.

as the decision interval of our scheme. We would also like to emphasize that we have compared with [7], which is an RF-based scheme in vehicular networks, which is regarded as state-of-the-art. We note that RF systems are the current technology being used to achieve low-latency communication in vehicular OCC networks.

B. Simulation Results

We start by exploring the training performance of our proposed actor-critic based DRL scheme by performing an ablation study to address the trade-off among different weight settings at the total rewards of (13), namely distance, ω_d; BER, ω_b; latency, ω_τ; and rate, ω_r, rewards. For ease of visual representation, we only demonstrate five particular settings, including (i) $\omega_d = 0.1$, $\omega_b = 0.1$, $\omega_\tau = 0.1$, $\omega_c = 0.7$; (ii) $\omega_d = 0.1$, $\omega_b = 0.4$, $\omega_\tau = 0.4$, $\omega_c = 0.1$; (iii) $\omega_d = 0.1$, $\omega_b = 0.5$, $\omega_\tau = 0.2$, $\omega_c = 0.2$; (iv) $\omega_d = 0.5$, $\omega_b = 0.2$, $\omega_\tau = 0.2$, $\omega_c = 0.1$; and (v) $\omega_d = 0.4$, $\omega_b = 0.1$, $\omega_\tau = 0.1$, $\omega_c = 0.4$, as shown in Fig. 3. We observe from the figure that setting (i) converges after 5000 decision timesteps and presents better loss performance when we assign higher weight related to throughput. Other settings have more elevated losses compared to setting (i). Though setting (ii) demonstrates better performance until 3000 timesteps, setting (i) overcomes setting (ii) after 3000 timesteps as the DRL agent takes some time to achieve balance between the action and the rewards. Therefore, we adopt this weights setting (i) throughout the performance evaluation.

To verify the improvement of rewards over decision interval, we illustrate the average rewards per 100 timesteps for the different variants of our proposed scheme, i.e., proposed scheme, greedy ($\zeta = 0$) and farsighted ($\zeta = 1$) for 10000 timesteps in Fig. 4. For the proposed scheme, we examine the achieved rewards for three different weight settings, i.e., (i), (iii) and (iv), as presented above, for the reward function. From the figure, we see that our proposed scheme with setting (i) demonstrates higher rewards over all the decision intervals,

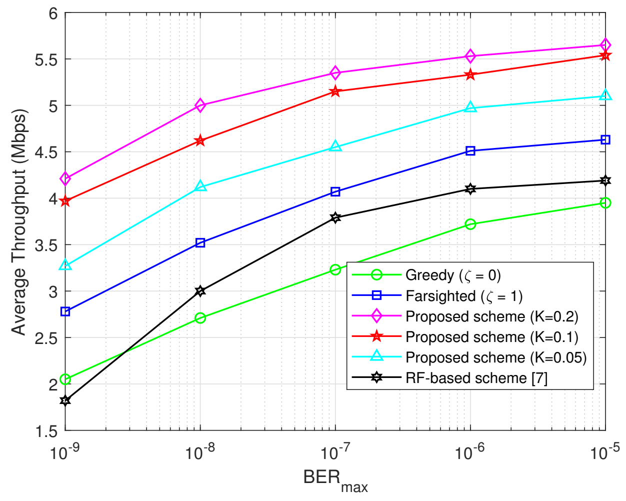


Fig. 5: Comparison of average rate by varying the BER$_{\max}$ requirement for all schemes under comparison.

whereas the rewards of greedy and farsighted schemes fluctuate. We also observe our proposed scheme with weights setting (i) reaches an average reward of 1 at 10000 timesteps, while the average reward for the other settings and variants of our scheme never reaches 1. This is because of the impact of weight settings on different sub-rewards. Since the weights setting (i) demonstrates better average reward performance than the other two weights settings, we adapt this setting for the rest of our simulations.

After demonstrating the training implementation, we now examine different communication performance metrics for our actor-critic-based vehicular OCC system. Consequently, we evaluate the schemes under comparison with respect to throughput, latency and reliability. Please note that for the RF-based scheme, we require communication between the server and agent back and forth through feedback link, which involves an extra delay that is usually 1 ms to 12 ms. For our simulations, we consider 2 ms as the additional delay for the feedback, which is a favourable setting for the RF scheme though other settings can also be used as user necessities.

First, in Fig. 5, we investigate the effect of BER on the average throughput (Mbps) for various BER$_{\max}$ values. From the figure, we see that the average throughput increases as we reduce the BER$_{\max}$ requirements from 10^{-9} to 10^{-5}. This happens because of using less strong LDPC codes. We observe that the proposed scheme with $K = 0.2$ achieves the highest average throughput for all BER values, whereas the performance of the proposed scheme with $K = 0.1$ is relatively close to the one achieved with $K = 0.2$ and the performance of the proposed scheme with $K = 0.05$ is close to Farsighted and significantly lower than our scheme with $K = 0.2$. This observation shows that increasing the number of neighbours can help improve the decision policy because the K value affects the number of actions that can be learned in a single iteration. We also notice that initially, the RF-based method achieves the lowest rate of all the schemes but performs better than the greedy method beyond BER$_{\max}$ = 10^{-8}. Hence, it is seen that when the BER requirements are tighter, other schemes fail to meet the constraint (6), and

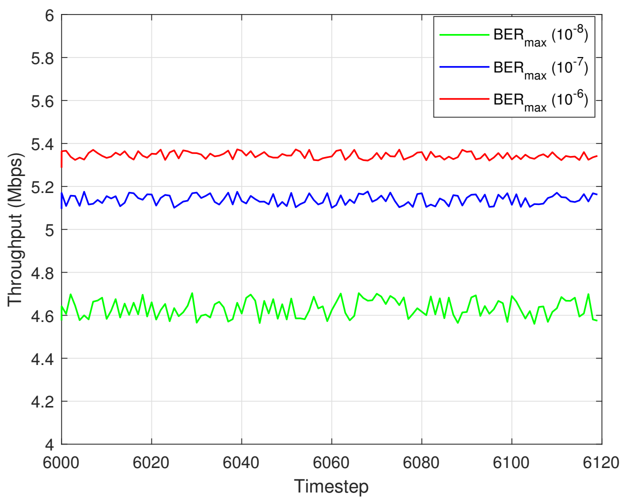


Fig. 6: Comparison of achievable throughput by our scheme per timestep considering different BER$_{\text{max}}$ requirement.

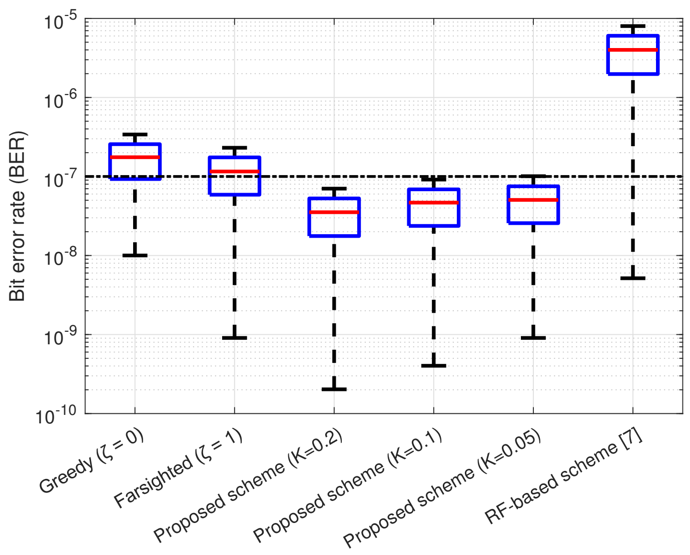


Fig. 7: Box plot to justify how the reliability requirement is satisfied considering our maximum allowable BER 10^{-7}.

therefore performance degrades. As a result, these schemes achieve lower average throughput compared to the proposed scheme at all times. Further, we would like to emphasise that the achieved average throughput by our proposed scheme for different BER requirements exceeds the minimum threshold required for seamless communication between vehicles which according to [52] is 4.1 Mbps. The work in [52] mentions that the required reliability is 10^{-4} which our system can always meet. This demonstrates that our approach maximizes the average throughput and meets the uRLLC requirements.

In an effort to present the robustness of selecting the code rate, we evaluate the throughput (Mbps) for three different BER$_{\text{max}}$ requirements in Fig. 6. For this simulation, we illustrate the throughput across 120 runs, where a sample is taken from the whole run at 6000 - 6120 timesteps. From the figure, we observe that the throughput varies near the average values for all BER$_{\text{max}}$. We also notice that we achieve higher throughput, and there is less fluctuation between decision intervals when the BER requirement is lower, e.g., 5.36 Mbps for 10^{-6}. At lower BER$_{\text{max}}$, the probability of violating (11) is less, and the throughput improves as the variation between one decision interval to another is shorter. At higher BER$_{\text{max}}$, the violation probability of (11) increases, and the throughput reduces for the bigger difference between one decision interval to another.

To visualize how the proposed scheme respects the uRLLC requirements while maximizing the throughput, we analyze the BER and latency performance for the various schemes under comparison. Please note that in this paper, we consider meeting BER of 10^{-7} and latency of 10 ms as the ultra-reliability and low-latency requirements, respectively. We execute the simulation for 10000 decision timesteps to investigate the BER and latency data. Again, we test the proposed framework with three different K values, i.e., $K = 0.2$, $K = 0.1$, and $K = 0.05$ and compare the BER and latency performance with that achieved by the other non-optimal schemes, greedy, farsighted, and RF-based scheme. We then generate boxplots over all the available data and compare the results with all the schemes under comparison. First, we illustrate the boxplot of the BER to demonstrate whether all the schemes under comparison meet the reliability requirement in Fig. 7. We have also plotted a reference line to present our BER constraint of 10^{-7} (dashed black line). The blue boxes in the boxplot represent the interquartile range, signifying the spread of the central 50% of the total values. The red lines inside the boxes denote the median, providing a measure of central tendency. Individual data points lying beyond the whiskers, which extend to the maximum and minimum values within the defined range, are considered potential outliers and are presented as dots.

Figure 7 shows that the proposed scheme satisfies the reliability requirement for all K values, whereas the other schemes fail to respect the constraint most of the time. We note that, the proposed scheme with $K = 0.2$ achieves the lowest BER than the other two K values of the proposed scheme and the proposed scheme with $K = 0.05$ has the highest BER performance. This is in accordance with what we have seen earlier, i.e., increasing the value of K can help an agent explore more action strategies and hence can achieve the lowest error rate. However, the performance with the higher K values is achieved at the cost of higher computation time [42].

Finally, we present the boxplot of observed latency for all comparison schemes in Fig. 8. Similar to the BER performance, we evaluate boxplots to examine how the low latency requirement (10 ms) is satisfied by the different comparison schemes. Like in the previous comparison, we also show a reference line for the latency constraint (dashed black line) in Fig. 8. The figure shows that the proposed scheme always respects the low latency requirements of 10 ms, while the other three schemes fail to meet the constraint most of the time in our simulation. In particular, for the greedy, farsighted and RF-based schemes, the maximum observed latency is 14.5 ms, 11.8 ms, and 17.5 ms, respectively. For the proposed scheme with different K values, the proposed scheme with the highest value, i.e., $K = 0.2$, has the lowest latency (1.5 ms to 8.7 ms),

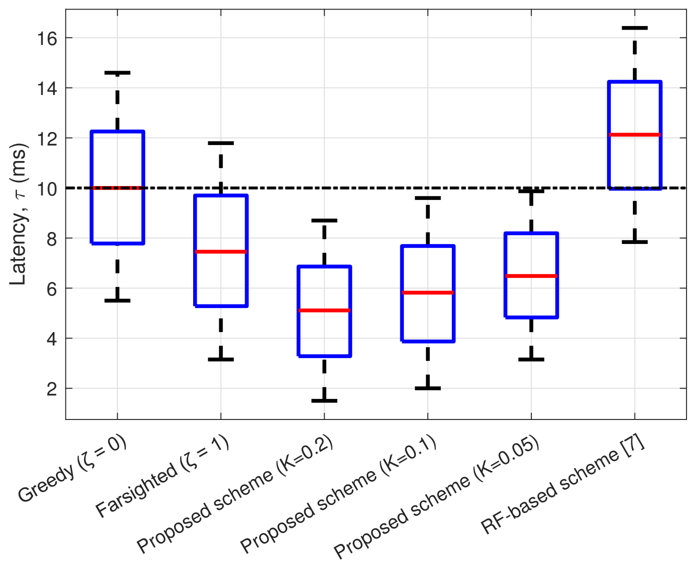


Fig. 8: Box plot to verify how the latency requirement is satisfied considering our latency requirement 10 ms.

whereas $K = 0.05$ offers latency in the range of 3.2 ms to 8.9 ms. The gap between the performance of different K values reflects the randomness that might be introduced by the proto action when considering different numbers of neighbours. From this comparison, we can summarize that the proposed vehicular OCC system can maximize the throughput while guaranteeing uRLLC, while the RF-based schemes cannot meet the delay requirements.

VIII. Conclusion

In this paper, we introduce an actor-critic DRL framework in vehicular OCC by selecting the optimal code rates and modulation schemes as well as changing the relative speed of the vehicles while respecting uRLLC requirements. In doing so, we first model the vehicular OCC system. To support variable rate and ultra-reliability, we use 5G NR LDPC code rate optimization for the M-QAM scheme. We solve the continuous optimization problem using an actor-critic algorithm with Wolpertinger architecture. We verify our proposed scheme through numerous simulations and compare it with several variants of our scheme and an RF communication-based scheme. The average throughput of our proposed scheme shows a considerably higher value compared to other schemes under comparison. We neglect the effect of weather conditions in this paper. Our proposed scheme can guarantee uRLLC while maximizing the throughput, whereas other methods fail most of the time. This happens because interference-free OCC DRL-based systems achieve higher rates even at low BER requirements, and the code rate optimization scheme offers ultra-reliability. We believe that our study is an important step toward guaranteeing uRLLC, and we hope that it will motivate researchers to conduct real-world experiments.

References

[1] S. Sun *et al.*, “Support for vehicle-to-everything services based on LTE,” *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 4–8, Jun. 2016.

[2] C. Liu and M. Bennis, “Ultra-reliable and low-latency vehicular transmission: An extreme value theory approach,” *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1292–1295, Jun. 2018.

[3] L. Zeng *et al.*, “High data rate multiple input multiple output (MIMO) optical wireless communications using white LED lighting,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 9, pp. 1654–1662, Dec. 2009.

[4] I. Takai *et al.*, “LED and CMOS image sensor based optical wireless communication system for automotive applications,” *IEEE Photon. J.*, vol. 5, no. 5, pp. 6 801 418–6 801 418, Oct. 2013.

[5] I. Takai *et al.*, “Optical vehicle-to-vehicle communication system using LED transmitter and camera receiver,” *IEEE Photon. J.*, vol. 6, no. 5, pp. 1–14, Oct. 2014.

[6] D. Hui, S. Sandberg, Y. Blankenship, M. Andersson, and L. Grosjean, “Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE,” *IEEE Veh. Technol. Mag.*, vol. 13, no. 4, pp. 60–69, Dec. 2018.

[7] 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, Aug. 2019.

[8] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, “Reinforcement learning for resource provisioning in the vehicular cloud,” *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 128–135, Jun. 2016.

[9] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*. MA, USA: MIT press Cambridge, 1998, vol. 135.

[10] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971v6*, Jul. 2019.

[11] G. Dulac-Arnold *et al.*, “Deep reinforcement learning in large discrete action spaces,” *arXiv preprint arXiv:1512.07679*, Apr. 2016.

[12] M. I. Ashraf, Chen-Feng Liu, M. Bennis, and W. Saad, “Towards low-latency and ultra-reliable vehicle-to-vehicle communication,” in *Proc. 2017 EuCNC*, Oulu, Finland, Jun. 2017, pp. 1–5.

[13] W. Sun, E. G. Ström, F. Brännström, Y. Sui, and K. C. Sou, “D2D-based V2V communications with latency and reliability constraints,” in *Proc. IEEE Globecom Workshops*, Austin, TX, Dec. 2014, pp. 1414–1419.

[14] M. M. K. Tareq, O. Semiari, M. A. Salehi, and W. Saad, “Ultra reliable, low latency vehicle-to-infrastructure wireless communications with edge computing,” in *Proc. 2018 IEEE GlobeCom*, Abu Dhabi, UAE, Dec. 2018, pp. 1–7.

[15] G. Vidyarthi, A. Ngom, and I. Stojmenovic, “A hybrid channel assignment approach using an efficient evolutionary strategy in wireless mobile networks,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1887–1895, Nov. 2005.

[16] A. Koller and S. D. Noble, “Domination analysis of greedy heuristics for the frequency assignment problem,” *Discrete Mathematics*, vol. 275, no. 1-3, pp. 331–338, Jan. 2004.

[17] Y. Goto, I. Takai, T. Yamazato, H. Okada, T. Fujii, S. Kawahito, S. Arai, T. Yendo, and K. Kamakura, “A new automotive VLC system using optical communication image sensor,” *IEEE Photon. J.*, vol. 8, no. 3, pp. 1–17, Jun. 2016.

[18] Z. Lu, C. Zhong, and M. C. Gursoy, “Dynamic channel access and power control in wireless interference networks via multi-agent deep reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1588–1601, Nov. 2021.

[19] S. Pfletschinger, A. Mourad, E. Lopez, D. Declercq, and G. Bacci, “Performance evaluation of non-binary LDPC codes on wireless channels,” in *Proc. ICT Mobile Summit*, Spain, Jun. 2009, pp. 1–8.

[20] M. Arabaci, I. B. Djordjevic, R. Saunders, and R. M. Marcoccia, “High-rate nonbinary regular quasi-cyclic LDPC codes for optical communications,” *J. Lightwave Technol.*, vol. 27, no. 23, pp. 5261–5267, Aug. 2009.

[21] A. Islam, M. T. Hossan, and Y. M. Jang, “Convolutional neural network scheme–based optical camera communication system for intelligent internet of vehicles,” *Int. J. Distrib. Sens. Netw.*, vol. 14, no. 4, pp. 1–15, Apr. 2018.

[22] P. Deng, “Real-time software-defined adaptive MIMO visible light communications,” *Visible Light Communications*, pp. 637–640, Jul. 2017.

[23] P. Luo *et al.*, “Experimental demonstration of a 1024-QAM optical camera communication system,” *IEEE Photon. Technol. Lett.*, vol. 28, no. 2, pp. 139–142, Oct. 2015.

[24] S. A. I. Alfarozi, K. Pasupa, H. Hashizume, K. Woraratpanya, and M. Sugimoto, “Square wave quadrature amplitude modulation for visible light communication using image sensor,” *IEEE Access*, vol. 7, pp. 94 806–94 821, Jul. 2019.

[25] A. Islam, N. Thomos, and L. Musavian, “Multi-agent deep reinforcement learning for spectral efficiency optimization in vehicular optical

camera communications," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 3666–3679, May 2024.

[26] C. Yoon, J.-E. Oh, M. Cheong, and S.-k. Lee, "A hardware efficient LDPC encoding scheme for exploiting decoder structure and resources," in *Proc. IEEE 65th VTC*, Dublin, Ireland, Apr. 2007, pp. 2445–2449.

[27] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[28] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, Mar. 2018.

[29] S. Seo, T. Mudge, Y. Zhu, and C. Chakrabarti, "Design and analysis of LDPC decoders for software defined radio," in *2007 IEEE Workshop on Signal Processing Systems*, Shanghai, China, Oct. 2007, pp. 210–215.

[30] K. Zhang, X. Huang, and Z. Wang, "High-throughput layered decoder implementation for quasi-cyclic LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 985–994, Jul. 2009.

[31] A. Islam, L. Musavian, and N. Thomos, "Performance analysis of vehicular optical camera communications: Roadmap to uRLLC," in *Proc. IEEE 2019 GLOBECOM*, Hawaii, USA, Dec. 2019, pp. 1–6.

[32] T. Yamazato *et al.*, "Vehicle motion and pixel illumination modeling for image sensor based visible light communication," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 9, pp. 1793–1805, Sep. 2015.

[33] D. A. Plaisted, "Some polynomial and integer divisibility problems are NP-HARD," in *Proc. 17th SFCS*, Houston, TX, USA, Oct. 1976, pp. 264–267.

[34] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*. Chelmsford, MA, USA: Courier Corporation, 1998.

[35] T. Zinchenko, "Reliability assessment of vehicle-to-vehicle communication," Doctoral Thesis, Carl Friedrich Gauss Faculty, Technische Universität Braunschweig, 2014.

[36] R. Wang, D. P. Foster, and S. M. Kakade, "What are the statistical limits of offline RL with linear function approximation?" *arXiv preprint arXiv:2010.11895*, 2020.

[37] X. Ma *et al.*, "Distributionally robust offline reinforcement learning with linear function approximation," *arXiv preprint arXiv:2209.06620*, 2022.

[38] D. Silver *et al.*, "Deterministic policy gradient algorithms," in *Proc. ICML*, Beijing, China, Jun. 2014, pp. 387–395.

[39] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, May 2014.

[40] V. Mnih *et al.*, "Playing Atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[41] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the Brownian motion," *Phys. Rev.*, vol. 36, no. 5, pp. 823–841, Sep. 1930.

[42] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep reinforcement learning-based edge caching in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 48–61, Jan. 2020.

[43] N. Papernot and P. McDaniel, "Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning," *arXiv preprint arXiv:1803.04765*, 2018.

[44] D. A. Guastella and G. Bontempi, "Traffic modeling with SUMO: A tutorial," *arXiv preprint arXiv:2304.05982*, 2023.

[45] J. H. Bae, A. Abotabl, H.-P. Lin, K.-B. Song, and J. Lee, "An overview of channel coding for 5G NR cellular communications," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–14, Jun. 2019.

[46] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[47] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Sep. 2018.

[48] 3GPP, "Study on scenarios and requirements for next generation access technologies," *Technical Specification Group Radio Access Network, Technical Report 38.913*, 2016.

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

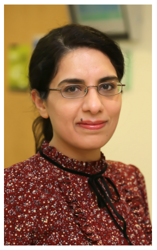
[50] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. on OSDI*, Savannah, GA, Nov. 2016, pp. 265–283.

[51] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2020.

[52] B. Lu, H. Zhang, T. Xue, S. Guo, and H. Gai, "Deep reinforcement learning-based power allocation for ultra reliable low latency communications in vehicular networks," in *Proc. 2021 IEEE/CIC ICC*, Xiamen, China, Jul. 2021, pp. 1149–1154.

Amirul Islam received his PhD degree in Computing and Electronic Systems from the University of Essex, UK, in 2022. He completed his MSc from Kookmin University, South Korea. He currently serves as an Assistant Professor in the Department of Electrical and Electronic Engineering at the American International University-Bangladesh, Bangladesh. Prior to this, he held the position of a Post-Doctoral Researcher at the Visual Artificial Intelligence Laboratory, Oxford Brookes University, UK. His research interests include machine learning for communication, optical camera communication, deep reinforcement learning, automotive vehicular communications, and optimization strategies.

Nikolaos Thomos (Senior Member, IEEE) is currently a Professor at the University of Essex, U.K. Before that, he was a Senior Researcher at the Ecole Polytechnique Federale de Lausanne (EPFL) and the University of Bern, Switzerland. He received the Diploma (MSc equivalent) and Ph.D. degrees from the Aristotle University of Thessaloniki, Greece. He has been PI and co-I for a number of ESPRC, SNSF, EC, InnovateUK, and Hassler foundation-funded projects. His research interests include machine learning for communications, multimedia communications, semantic communications, design of new waveforms, cross-layer optimization, network coding, information-centric networking, source and channel coding, and signal processing. He is an elected member of the IEEE MMSP Technical Committee (MMSP-TC) for the period 2019-2024. He received the highly esteemed Ambizione Career Award from the Swiss National Science Foundation (SNSF).



Leila Musavian is a Professor at University of Essex. Previously, she was Deputy Pro-Vice-Chancellor for Research at University of Essex and Reader in Telecommunications at the School of Computer Science and Electronic Engineering. Prior to that, she was Lecturer/Senior Lecturer at InfoLab21, Lancaster University, UK, Research Associate at McGill University, Canada, Research Associate at Loughborough University, UK and Post-Doctoral Fellow at INRS-EMT, University of Quebec, Canada.

Her research interests lie in Radio Resource Management for 6G/B5G communications, low latency communications, ML for Communications, NFC, NTN and Energy Harvesting Communications. She is currently editor of the IEEE Communications Surveys and Tutorials (COMST) and has previously been Editor of IEEE TRANSACTIONS OF WIRELESS COMMUNICATIONS. She was the Conference Workshop Co-Chair of VTC-Spring-2020, the Wireless Communications Symposium Leading Co-Chair for IEEE ICC 2021, Montreal, Canada, conference TPC Co-Chair of IEEE CAMAD 2021, Portugal and Track 2 Lead chair of IEEE WCNC 2023. She is the founding chair and co-chair of the IEEE UK & Ireland Future Networks local group.