


Article

Ultra-Reliable Deep-Reinforcement-Learning-Based Intelligent Downlink Scheduling for 5G New Radio-Vehicle to Infrastructure Scenarios

Jizhe Wang¹, Yuanbing Zheng¹, Jian Wang¹, Zhenghua Shen¹, Lei Tong¹, Yahao Jing², Yu Luo² and Yong Liao^{2,*} 

¹ State Grid Chongqing Information and Telecommunication Company, Chongqing 400012, China

² School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

* Correspondence: liaoy@cqu.edu.cn

Abstract: Higher standards for reliability and efficiency apply to the connection between vehicle terminals and infrastructure by the fifth-generation mobile communication technology (5G). A vehicle-to-infrastructure system uses a communication system called NR-V2I (New Radio-Vehicle to Infrastructure), which uses Link Adaptation (LA) technology to communicate in constantly changing V2I to increase the efficacy and reliability of V2I information transmission. This paper proposes a Double Deep Q-learning (DDQL) LA scheduling algorithm for optimizing the modulation and coding scheme (MCS) of autonomous driving vehicles in V2I communication. The problem with the Doppler shift and complex fast time-varying channels reducing the reliability of information transmission in V2I scenarios is that they make it less likely that the information will be transmitted accurately. Schedules for autonomous vehicles using Space Division Multiplexing (SDM) and MCS are used in V2I communications. To address the issue of Deep Q-learning (DQL) overestimation in the Q-Network learning process, the approach integrates Deep Neural Network (DNN) and Double Q-Network (DDQN). The findings of this study demonstrate that the suggested algorithm can adapt to complex channel environments with varying vehicle speeds in V2I scenarios and by choosing the best scheduling scheme for V2I road information transmission using a combination of MCS. SDM not only increases the accuracy of the transmission of road safety information but also helps to foster cooperation and communication between vehicle terminals to realize cooperative driving.

Keywords: 5G; NR-V2I; automatic driving; DDQL; ultra-reliable



Citation: Wang, J.; Zheng, Y.; Wang, J.; Shen, Z.; Tong, L.; Jing, Y.; Luo, Y.; Liao, Y. Ultra-Reliable Deep-Reinforcement-Learning-Based Intelligent Downlink Scheduling for 5G New Radio-Vehicle to Infrastructure Scenarios. *Sensors* **2023**, *23*, 8454. <https://doi.org/10.3390/s23208454>

Academic Editors: Chen Chen, Qingqi Pei, Kai Liu, Lei Liu and Dapeng Lan

Received: 15 August 2023
Revised: 17 September 2023
Accepted: 11 October 2023
Published: 13 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Vehicles with autonomous driving capabilities are presently advancing quite rapidly. Numerous developments and investigations have been conducted recently to enhance the capacity of connected automobiles to transmit data about their surroundings. The vehicles to everything (V2X) is a sizable interactive network made up of vehicle location information including speed and location, and it involves four different types of communication: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-network (V2N), and vehicle-to-pedestrian (V2P) [1]. The intelligent transportation system (ITS), which is intended to improve driving convenience and safety, includes V2I communication technology as a key component. Vehicles can receive more comprehensive road information from the infrastructure, like warnings about construction zones, traffic accidents, and traffic congestion, enabling them to make better driving judgments. In order to increase traffic efficiency and lessen congestion, the infrastructure can also alter the timing of signal lights and optimize the timing of traffic signals through communication with vehicles [2].

Users have extremely high expectations for ultra-reliable and low-latency communication (URLLC) in the V2I scenario of the Internet of vehicles, which is also essential for

maintaining road safety. The Internet of vehicles and fifth-generation mobile communication technology (5G) are both developing at the same time, and NR-V2X leverages link adaptation (LA) to give URLLC more robust technical support. Through the use of the V2I channel quality adjustment modulation and coding scheme (MCS) in NR-V2X, LA may provide reliable transmission. Adaptive modulation and coding (AMC) make it possible for ITS's intelligent vehicle communications to have better spectrum awareness ability [3].

The AMC modifies the transmission parameters in accordance with the channel's quality at each given time. Data transmission rates can increase with faster modulation and encoding rates. If the channel conditions are bad, some transmission rates can be sacrificed to lower transmission mistakes, while the modulation methods and coding rates can be decreased to retain reliability. Fixed lookup tables, inner loop link adaptation (ILLA), outer loop link adaptation (OLLA), and no outer loop link adaptation (NoOLLA) technologies are common components of traditional AMC solutions. In the realm of AMC, OLLA technology is a higher-level adaptive technique that has the ability to dynamically modify the settings in accordance with network resources and global performance indicators. While using a predefined parameter configuration for data transmission, NoOLLA technology is a fixed method that is easier and does not require the idea of outer ring adjustment [4]. The first receiver provides feedback on the channel state information (CSI) in the conventional AMC. The transmitter then examines the channel state data to determine the correlation between the channel quality index (CQI) and the signal-to-noise ratio (SNR). The transmitter will automatically modify MCS to achieve adaptive switching based on this relationship [5]. In a V2I scenario, the vehicle's fluctuating speed and the random scattering phenomenon in a high-speed driving environment would cause the transmission signal to travel along a number of different paths as it attempts to reach the base station (BS). Due to the separate and quick temporal phase shifts caused by the various Doppler shift on these paths, the channel rapidly fades (for instance, the amplitude and phase of the entire channel change quickly over time). In this instance, a channel quality indicator based solely on SNR has been unable to adequately depict the channel's actual state. The effectiveness of communicating information about road safety and the throughput of data communication may suffer significantly as a result of the effects of rapid deterioration [6].

The use of machine learning (ML) technology in ITS has grown significantly in recent years [7]. The literature [8,9] discusses the use of deep learning algorithms in AMC and compares the effectiveness of algorithms, such as convolutional neural network (CNN), ResNet, DenseNet, and convolutional long and short-term deep neural network (CLDNN), in classifying signal modulation types. The ML technique of reinforcement learning (RL) has also been used for a variety of issues, such as resource optimization, coverage and capacity optimization, and backhaul optimization [10]. According to the literature [11], when using RL in AMC, the received signal to interference-plus-noise ratio (SINR) is used to determine the MCS, and because SINR is a continuous variable, the state space is similarly continuous. When dealing with such a continuous state space, this enables the learning algorithm to take a wider state space into consideration. According to the literature [12], the MCS selection rules are modified using RL algorithms in order to take into account the consequences of prior AMC judgments. According to the literature [4], based on the Q learning algorithm, BS can independently investigate and choose the best MCS schemes to maximize spectral efficiency while retaining a low bit error rate (BER). In order to help agents deal with high-dimensional state spaces, learn complex strategies, increase learning efficiency, and apply to the continuous motion space problem, deep reinforcement learning (DRL) combines the benefits of deep learning and RL [13]. Based on this, a study [14] utilizing DRL developed an intelligent MCS selection algorithm with outstanding transmission rate performance in the setting of cognitive heterogeneous networks. The Deep Q-network (DQN) algorithm is a popular one for DRL. For the joint scheduling of MCS and space division multiplexing (SDM) in the 5G massive MIMO-OFDM system, the literature [15] suggests a DQN-based approach.

Traditional DQN uses a single neural network for both action selection and Q value estimation, which leads to an excessive Q value estimate [15]. Two neural networks are introduced by the double deep Q-network (DDQN), one for action selection and the other for Q value estimation [16]. By choosing an action and assessing its Q value at each update, this dual-network structure can decrease the overestimation of Q value and improve the stability and performance of DDQN [17]. Therefore, in order to improve the performance of the DQN-based scheduling algorithm in the literature [14] and make it more adapted to ultra-reliable intelligent downlink scheduling, this paper suggests a massive MIMO intelligent scheduling technique based on DDQN for the 5G NR-V2I scenario. This approach is employed for intelligent joint scheduling of MCS, precoding matrix indicator (PMI), and SDM. This paper suggests a highly trustworthy intelligent downlink scheduling technique based on DDQN for the 5G NR-V2I scenario. The following are its specific contributions:

- (1) Eliminate the overvaluation issue with Q value—when learning the Q value function for the DQN algorithm, the Q value is prone to being overstated, which means that for some state–action combinations, its Q value might overestimate. Due to this, the DQN algorithm may occasionally choose ineffective actions, which will have an impact on the scheduling efficiency. The overestimation problem of Q values can be reduced by DDQN by using two Q networks, one for choosing actions and the other for assessing the value of those activities, therefore enhancing the precision and stability of downlink scheduling algorithm learning.
- (2) More precise action choice—dual Q networks are utilized by the DDQN algorithm to pick activities, which allows for a more precise assessment of the relative worth of various actions. Due to this, DDQN may be able to choose actions with greater precision, improving the downlink scheduling approach. The DDQN algorithm can more precisely choose the actions that can optimize throughput or lower the BER, thereby enhancing link performance, when compared to DQN, OLLA, and NoOLLA.
- (3) Overcoming the issue of the local optimal solution—the OLLA algorithm may enter the local optimal solution and fail to attain the global optimal by optimizing the local action selection. The DDQN algorithm, in contrast, employs dual Q networks throughout the learning phase, which can better avoid the local optimal solution problem and more effectively explore the larger action space.
- (4) Adapt to surroundings that are more complicated—by using two Q networks and reinforcement learning, the DDQN algorithm can adapt more flexibly to various channel environments and network requirements under a dynamic, changing environment, so as to improve the efficiency and reliability of communication links. This makes DDQN have strong adaptability and superior performance in a complex environment.

This paper is organized as follows. The downlink adaptive scheduling model based on the channel-state information reference signal (CSI-RS) is primarily established in Section 2. The adaptive technique of V2I downlink scheduling based on DDQN is introduced in Section 3, along with the measurement of the downlink channel, data processing, network architecture, and training parameter setup. In Section 4, the simulation results are verified. The conclusion is provided in Section 5.

2. Problem Formulation

Through the policy modification of the downlink communication, NR-V2I improves the communication reliability and spectrum efficiency of the vehicle terminal. The application scenario of NR-V2I [18] is given in Figure 1. A lower modulation scheme and coding rate can be utilized when the edge-Internet of vehicles (E-IoV) server delivers signals to the vehicle terminal through the road side unit (RSU), which will boost the robustness for weak connections. In addition, the E-IoV Server increases spectral efficiency (SE) by using a higher modulation scheme and coding rate.

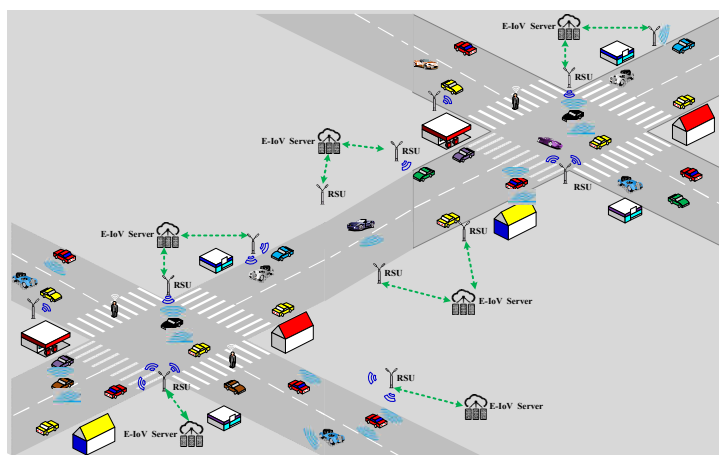


Figure 1. Communication scenarios for NR-V2I.

The MIMO-OFDM communication system of NR-V2I [19] (Individual User 1) is used as the research subject in this work. The intelligent link scheduling approach based on DRL is used in the downlink adaptive scheduling of CSI-RS. In Figure 2, the scheduling is displayed. The fundamental principles of NR-V2I communication are as follows.

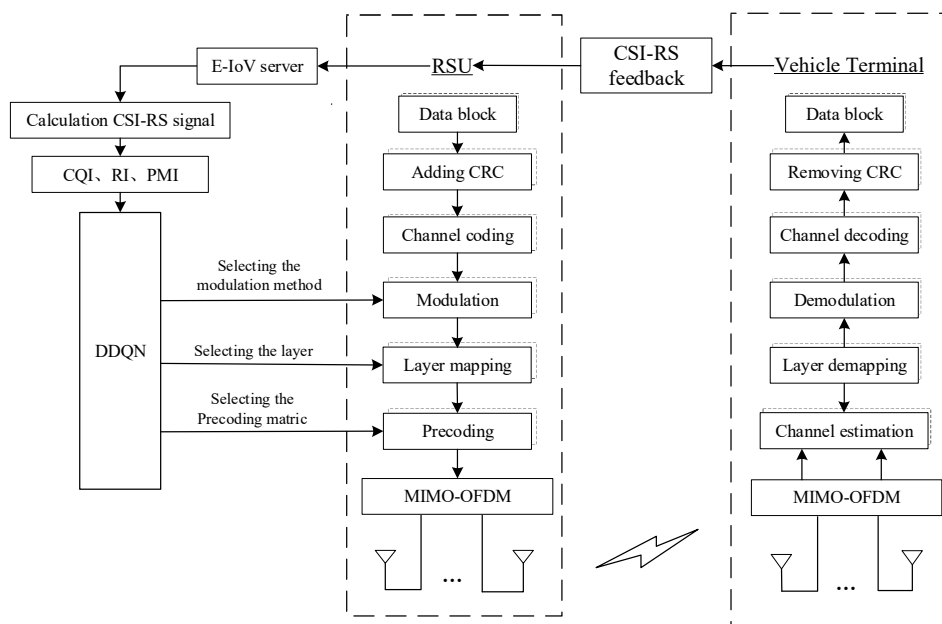


Figure 2. Scheduling for NR-V2I communication systems with reliable links.

The vehicle terminal measures the CSI-RS sent from the RSU side and then feeds the signal back to the RSU through the physical uplink data channel. The E-IoV server chooses the downlink scheduling scheme based on the feedback value of the CSI-RS transmitted by the RSU, which provides an ultra-reliable and low-latency communication scheme for the current data transmission of in-vehicle terminals through the DDQN method.

Consider how [20] may be employed to describe the channel capacity in a MIMO context.

$$V(H) = \log_2 \left\{ \det \left[E_{N_r} + \eta(WH)(WH)^H \right] \right\} \tag{1}$$

where V is the channel capacity; $H \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix; N_t and N_r are, respectively, the number of transmitting and receiving antennas; the letter E_{N_r} stands for the unit matrix in N_r dimensions; η denotes the signal transmitting power to noise power ratio; W is the beam fugitive matrix; $(\cdot)^H$ indicates the conjugate transpose matrix of the solver matrix;

and $\det(\cdot)$ is the solver matrix's determinant. The RSU's downlink adaptive scheduling, which is closely connected to the RSU's downlink adaptive scheduling, has a significant impact on the BER of the real downlink of the NR-V2I communication system.

The code elements in the NR-V2I communication system are encoded in an OFDM resource block (RB) for cyclic redundancy check (CRC), and if the check is unsuccessful, all of the RB's code elements are retransmitted. You may obtain the downlink BER B_{slot} for a single time slot by:

$$B_{\text{slot}} = B_e / (l \cdot c \cdot m \cdot N_{\text{RB}} \cdot N_{\text{RE}}) \quad (2)$$

where B_e refers to the number of downlink transmission error bits; l is the number of downlink-scheduled layers for air-division multiplexing; c is the number of downlink-scheduled bits for data transmission code; m indicates the number of modulated downlink-scheduled data symbols; N_{RB} denotes the number of downlink-scheduled resource blocks (RBs); and N_{RE} is in the name of the number of resource blocks (Res) that make up each RB. When the subcarrier spacing is 15 kHz, there are 14 OFDM symbols and 12 subcarriers in one RB in OFDM. N_{RB} and N_{RE} are treated as fixed values in this paper. They primarily depend on the resource allocation and are independent of the link-adaptive downlink scheduling policy.

A mathematical description of the downlink adaptive scheduling method based on the CSI-RS may be obtained from (3):

$$\underset{B_e, P_{\text{SE}}}{\text{argmin}} B_{\text{slot}} = B_e / (P_{\text{SE}} N_{\text{RB}} N_{\text{RE}}) \quad (3)$$

$$\text{s.t. } P_{\text{SE}} = l \cdot P_{\text{U-SE}} = f(D_{\text{CQI}}, D_{\text{RI}}, D_{\text{PMI}}, B_{\text{P-slot}}) \quad (3a)$$

$$1 \leq l \leq 4 \quad (3b)$$

$$P_{\text{U-SE}} = c \cdot m = M(D_{\text{MCS}}) \quad (3c)$$

$$m \in \{1, 4, 6, 8\} \quad (3d)$$

The intention of the downlink adaptation based on the CSI-RS is to reduce the BER. B_{slot} represents the number of incorrect bits following the current time slot scheduling. The state variables are the CQI, RI and PMI determined by the E-IoV Server based on the CSI-RS fed back from the vehicle terminals delivered by the RSU and the BER $B_{\text{P-slot}}$ obtained through statistics after the prior time slot has been scheduled. The decision variables l are and D_{MCS} .

$$P_{\text{SE}} = r \cdot P_{\text{U-SE}} = l \cdot M(D_{\text{MCS}}) = f(D_{\text{CQI}}, D_{\text{RI}}, D_{\text{PMI}}, B_{\text{P-slot}}) \quad (4)$$

where, as indicated in Equation (4), the spectral efficiency is P_{SE} . Furthermore, the D_{CQI} , D_{RI} and D_{PMI} stand for, respectively, the CQI, RI and PMI calculated by the E-IoV server. $f(\cdot)$ stands for the downlink adaptive scheduling algorithm based on the CSI-RS, with the SEs discounted by the l and D_{MCS} as their outputs. The algorithm's inputs are the CQI, RI, PMI and B_e supplied by the E-IoV server.

In Equation (5), $P_{\text{U-SE}}$ stands for Unit-Spectral Efficiency, or U-SE.

$$P_{\text{U-SE}} = l \cdot m = M(D_{\text{MCS}}) \quad (5)$$

where the $M(D_{\text{MCS}})$ function represents the U-SE acquired at a certain order D_{MCS} that corresponds to the current order. The primary scheduling parameters produced by the downlink adaptive method are the number of downlink air-division multiplexing layers l , the downlink data coding rate c , and the downlink symbol modulation order m . $(c \cdot m)$ symbolizes the number of bits that are acceptable on a single RE.

When the scheduling of l and D_{MCS} grows more than the current channel conditions of the vehicle terminal support demodulation capacity, B_e and B_{slot} shall grow. The downlink space division multiplexing layer number l and MCS order D_{MCS} two parameters primarily reflect the transmission data density. In addition, even when B_e is reduced, the system's B_{slot} will not reach the minimum value of B_{slot} due to the excessively conservative amount of scheduling data when l and D_{MCS} scheduling tend to be significantly less than the demodulation capability supported by the vehicle terminal under the current channel conditions. With the goal of bringing the system into balance with the B_e while minimizing the system's B_{slot} , the number of layers l of downlink space division multiplexing and the order of MCS D_{MCS} scheduling must be closely matched to the current channel state and the demodulation capability of the vehicle terminals.

3. DDQN-Based V2I Downlink Scheduling Adaptation

3.1. Downlink Channel Measurement

For the purpose of downlink channel measurement in the NR-V2I communication system depicted in Figure 2, the RSU periodically inserts the CSI-RS into the downlink data frame and then transmits it to the onboard terminal. The scheduling strategy for the downlink will ultimately be influenced by the measuring results of the feedback from the onboard terminal to the RSU. If the onboard terminal has N_r receiving antennas and N_t transmitting antennas at the RSU, and the signal flow during transmission is described as

$$y_{\text{CSI-RS}} = S_{\text{CSI-RS}}h_{\text{DL}} + n_{\text{DL}} \tag{6}$$

the remaining ports transmit zero pilot because the CSI-RS is mapped to various time-frequency domain positions on various transmitting antennas. We can therefore infer that CSI-RS per transmitting antenna is:

$$S_{\text{CSI-RS}} = \text{diag}(s_{\text{CSI-RS}}) \tag{7}$$

The emitted CSI-RS vector can be expressed as $s_{\text{CSI-RS}} = [s_1 \ s_2 \ \dots \ s_q \ \dots \ s_{N_r}]^T$ because $\text{diag}(\cdot)$ indicates building $s_{\text{CSI-RS}}$ as a diagonal matrix. The CSI-RS vector of each transmitting antenna to the receiving antenna q may be expressed as $s_q = [s_{q,1} \ s_{q,2} \ \dots \ s_{q,p} \ \dots \ s_{q,N_t}]$.

The received CSI-RS vector is expressed as $y_{\text{CSI-RS}} = [y_1^T \ y_2^T \ \dots \ y_q^T \ \dots \ y_{N_r}^T]^T$; however, the CSI-RS vector received by the receiving antenna may be expressed as $y_q = [y_{q,1} \ y_{q,2} \ \dots \ y_{q,p} \ \dots \ y_{q,N_t}]$. Additionally, the channel response on the receiving antenna q is represented as $h_q = [h_{q,1} \ h_{q,2} \ \dots \ h_{q,p} \ \dots \ h_{q,N_t}]$; hence, the downstream channel's channel response is $h_{\text{DL}} = [h_1^T \ h_2^T \ \dots \ h_q^T \ \dots \ h_{N_r}^T]^T$. A noise vector δ_n^2 with a mean of 0 and a variance of $n_{\text{DL}} \in \mathbb{C}^{N_t N_r \times 1}$ is then used to represent the noise on the channel.

Formula (6) and the CSI-RS of each transmitting antenna allow for the least square (LS) estimation of the downstream channel response vector \hat{h}_{DL} :

$$\hat{h}_{\text{DL}} = (S_{\text{CSI-RS}})^{-1}y_{\text{CSI-RS}} \tag{8}$$

Additionally, obtain the downlink channel response matrix \hat{H}_{DL} :

$$\hat{H}_{\text{DL}} = \begin{bmatrix} \hat{h}_1^T & \hat{h}_2^T & \dots & \hat{h}_{N_r}^T \end{bmatrix} = \begin{bmatrix} \hat{h}_{1,1} & \hat{h}_{2,1} & \dots & \hat{h}_{N_r,1} \\ \hat{h}_{1,2} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \hat{h}_{1,N_t} & \dots & \dots & \hat{h}_{N_r,N_t} \end{bmatrix} \tag{9}$$

Then, the onboard terminal will obtain the RI, PMI, and CQI based on the estimated \hat{H}_{DL} measurement and feed the above measurements back to the RSU. The vehicle terminal will be based on the estimated \hat{H}_{DL} . The value of RI is usually related to the number of antennas and the channel environment, and higher RI values indicate better space fraction multiplexing capability. The eigenvalues of the channel matrix are obtained by performing an eigenvalue decomposition of the channel matrix \hat{H}_{DL} .

$$\hat{H}_{DL} = U_{DL} \Sigma_{DL} V_{DL}^H \quad (10)$$

In particular, the eigenvalues Σ_{DL} reflect the singular values of the channel, which reflect the capacity of the channel to transmit signals across its many layers. U_{DL} and V_{DL} are unitary matrices. Consequently, the RI can be determined using the following equation:

$$D_{RI} = \begin{cases} 0 & \text{if } Z(\Sigma_{DL} - N_{DL}) = 0, 1 \\ Z(\Sigma_{DL} - N_{DL}) & \text{else} \end{cases} \quad (11)$$

The $Z(\cdot)$ function determines the number of diagonal elements in the matrix that are greater than zero, where $N_{DL} = \delta_n^2 I_{N_r}$ is the noise matrix of each layer.

When RI values are known, they can be mapped to the corresponding precoded matrix index using predefined PMI tables [21]. The collection of possible PMI matrices is \mathbb{S}_{PMI} , and the values of N_t , N_r and D_{RI} are known. If the PMI matrix corresponding to the PMI matrix index D_{PMI} is $W_{D_{PMI}}$, $W_{D_{PMI}} \in \mathbb{S}_{PMI}$, it will assume that element \mathbb{S}_{PMI} has N_{PMI} elements. The estimated SNR matrix $\Gamma_{D_{PMI}}$ can be computed using the downlink precoding matrix $W_{D_{PMI}}$, as follows:

$$\Gamma_{i_{PMI}} = \left[N_{DL} \left(W_{D_{PMI}} \hat{H}_{DL}^H \hat{H}_{DL} W_{D_{PMI}}^H + N_{DL} \right)^{-1} \right]^{-1} \quad (12)$$

To fully account for the influences between multiple levels, the SNRs of each layer were merged to obtain an integrated SNR value. A second norm of $\Gamma_{D_{PMI}}$ can be used to produce the combined SNR ρ_{PMI} . There are various PMI matrices available in the collection of PMIs, each of which corresponds to a distinct precoding technique. Because there are fewer aggregate elements, the onboard terminal can poll (or traverse) each PMI index in turn and determine the appropriate combined SNR value. The merged SNR value for each candidate PMI index was calculated, and the PMI index that maximizes the SNR value was then identified. The onboard terminal returns the index to the RSU in the following manner after locating the ideal PMI index:

$$\operatorname{argmax}_{D_{PMI}} \rho_{PMI} = \|\Gamma_{D_{PMI}}\|_F^2 \quad (13)$$

The CQI is a channel quality indicator that is frequently used in communication systems for adaptive modulation and encoding [22]. The mapping function D_{CQI} can be used to determine the appropriate CQI index $M_{CQI}(\cdot)$ for the decibel representation of ρ_{PMI} :

$$D_{CQI} = M_{CQI}(\log_2(\rho_{PMI} - 1)) \quad (14)$$

The onboard terminal will now encode the RI, PMI, and CQI measured data into a feedback signal and transmit them back to the RSU. The RSU will decide the downlink scheduling choice method based on the aforementioned facts after receiving this report.

3.2. Data Processing and Network Architecture

The direct application of the DQN algorithm will end up resulting in an overestimation of the decision value [23] due to the complexity of the NR-V2I communication system, the analog nature of the states and actions, and the volume of data. As a result, in this paper,

we use the DDQN for downlink scheduling and the DNN network for calculating the Q value rather than the Q-Table. Figure 3 depicts the DDQN's structural layout.

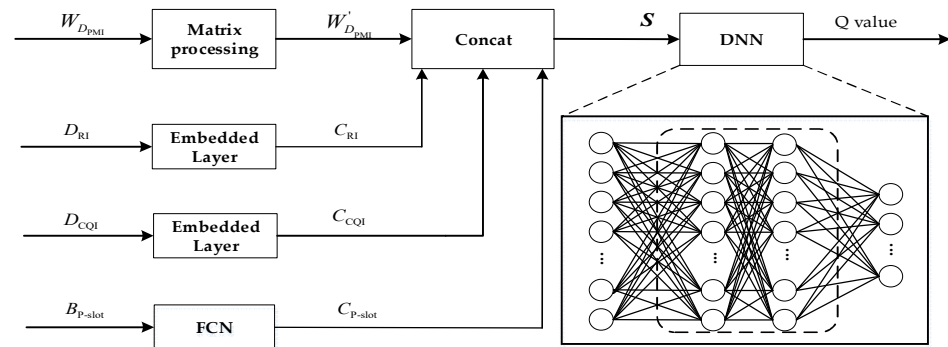


Figure 3. Schematic diagram of DDQN network structure.

Figure 3 depicts the network structure used in DDQN. It primarily consists of the data preprocessing section, the Concat layer, and the DNN layer. The DNN layer is a Full Convolutional Neural Network (FCN), where the input is the current state S and the output is the Q value of the reward value corresponding to all of the actions in the current state.

The D_{CQI} , D_{RI} and D_{PMI} from the measurement feedback of the vehicle terminal, as well as the statistically obtained B_{P-slot} , are the primary sources of information for the DDQN used in this paper to output downlink adaptive scheduling. Because the dimensionality of each variable varies, it is necessary to preprocess the data before inputting them into the DNN network. The preprocessing of input data to the DNN network consists of the following parts:

1. Matrix processing: Equation (15) illustrates how one may acquire the precoding matrix $W_{D_{PMI}} \in \mathbb{C}^{N_t \times N_r}$ for the precoding matrix and obtain $W'_{D_{PMI}} \in \mathbb{R}^{2N_t \times N_r}$ following the same matrix processing:

$$W'_{D_{PMI}} = \begin{bmatrix} \text{Re}(W_{D_{PMI}}) \\ \text{Im}(W_{D_{PMI}}) \end{bmatrix} \quad (15)$$

where $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ are shown as taking, respectively, the real part function and the imagistic part function.

2. Embedding layer: As a result of $D_{CQI} = (0, 15) \in Z$, $D_{RI} = (0, 3) \in Z$, the CQI encoding vector and RI encoding vector must be obtained to satisfy the network input conditions. These vectors can be obtained by the embedding layer network in deep learning, and the embedding layer can transform the input's index value into a vector of a specific dimension size. The embedding layer, in particular, is essentially made up of several fully connected networks, but it has a different focus. The output of the embedding layer is equivalent to the weights in the fully connected network, which acquires the network weights.

Given that there are 16 and 4 CQI and RI values in this research, respectively, and that each coding vector possesses a dimension of N_r , the embedding matrix may be represented as follows:

$$E_{CQI} = [e_1^{CQI}, e_2^{CQI}, \dots, e_{16}^{CQI}]^T \quad (16)$$

$$E_{RI} = [e_1^{RI}, e_2^{RI}, \dots, e_4^{RI}]^T \quad (17)$$

where $E_{CQI} \in \mathbb{R}^{16 \times N_r}$ and $E_{RI} \in \mathbb{R}^{4 \times N_r}$. Before training, the data in the embedding matrix are set up at random. During training, the embedding layer can obtain the specified row

vector in the embedding matrix as the coding vector according to the given input index value by simply applying the values of D_{CQI} and D_{RI} , whose expressions are, respectively:

$$C_{CQI} = S(E_{CQI}, D_{CQI}) \quad (18)$$

$$C_{RI} = S(E_{RI}, D_{RI}) \quad (19)$$

where $D_{CQI} \in \mathbb{R}^{1 \times N_r}$ and $C_{RI} \in \mathbb{R}^{1 \times N_r}$ are the CQI encoding vector and the RI encoding vector under the input D_{CQI} and D_{RI} values, respectively; $S(\cdot)$ indicates that the specified row vector in the matrix is picked as the encoding vector based on the index value.

3. Fully Connected Layer: To be able to obtain the mapping vector $C_{P\text{-slot}} \in \mathbb{R}^{1 \times N_r}$ of BER, high-dimensional mapping will be executed by applying the FCN network's $B_{P\text{-slot}}$ because $B_{P\text{-slot}} = (0, 1) \in Q$.
4. Concat operation: Following the previously mentioned process, the processed data must be concatenated into a single dimension to receive the DNN layer's input.

$$S = \text{Concat} \begin{pmatrix} W'_{D_{PMI}} \\ D_{CQI} \\ D_{RI} \\ D_{P\text{-slot}} \end{pmatrix} \in \mathbb{R}^{(3+2N_r) \times N_r} \quad (20)$$

where $\text{Concat}(\cdot)$ denotes the splicing function and S is the input to the DNN layer.

In this paper, the basic elements of the Q-learning algorithm in a DDQN system are represented as:

- (1) Environment (environment): communication system with adaptive scheduling for NR-V2I downlink;
- (2) Intelligent body (agent): vehicle-mounted terminal;
- (3) Action: the quantity of space division multiplexing layers RI and MCS used by downlink scheduling by RSU, which is referred to as action $a = (r, D_{MCS})$ in DDQN;
- (4) State: states are defined as those that are explicitly specified, as indicated in Equation (20), such as the D_{CQI} acquired from downlink measurement, the precoding matrix $W_{D_{PMI}}$ corresponding to D_{PMI} and D_{RI} , and the state matrix S produced from $B_{P\text{-slot}}$ after data preprocessing;
- (5) Reward: B , which is specified as indicated in Equation (5), is defined as the BER following downlink adaptive scheduling.

A neural network is utilized to estimate the Q value rather than a Q-Table in the downlink scheduling technique based on DDQN, which was created by fusing the DNN network illustrated in Figure 4 with the Q-learning algorithm. The problem of overestimation in DQN is resolved by the reinforcement learning technique known as DDQN by splitting the computation of the desired Q value into two steps: action selection and value evaluation. The overestimation issue in DQN is resolved by DDQN, a reinforcement learning technique, by splitting the computation of target Q values into two steps: action selection and value evaluation. A memory database is inherited by the DQN to solve the relevance problem of consecutive samples. The memory database stores past experiences, such as a specific number of (state, action, reward, and next state) sample data acquired in the setting of the NR-V2I communication system, and it randomly selects a small batch of sample data to train the network in the training phase. This enables a more effective training of the DNN by using both the old and new data.

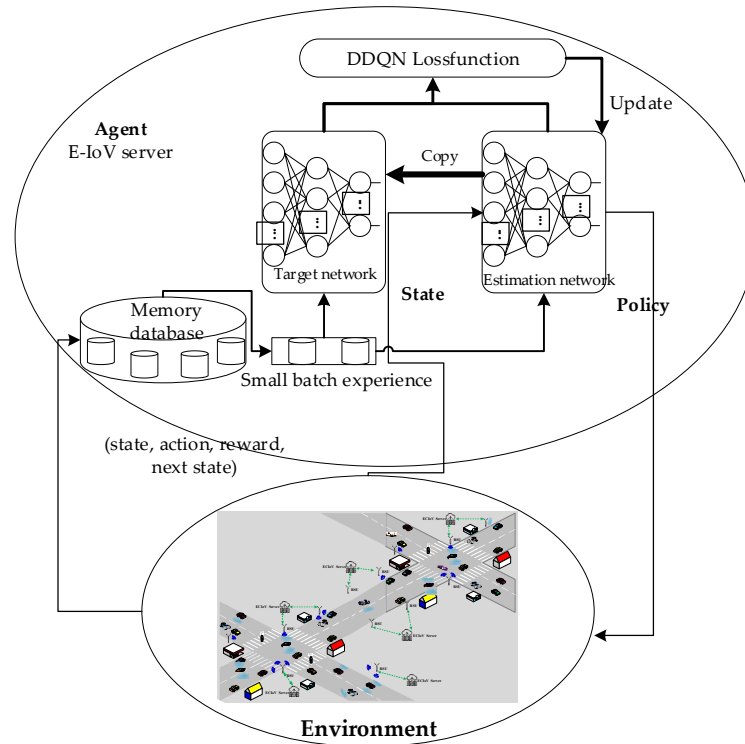


Figure 4. Reliable link scheduling structure based on DDQN.

A nonlinear approach is used in DDQN to represent the Q estimator function $Q(S, a; \theta)$, where θ is a parameter of the neural network, and then the loss function in the DNN network is defined as:

$$L(\theta) = E[Q^+(S, a) - Q(S, a; \theta)]^2 \quad (21)$$

The parameter update of the neural network can be expressed as:

$$\theta^{(i+1)} \leftarrow \theta^{(i)} - \delta \nabla L(\theta^{(i)}) \quad (22)$$

Both the computational network and the target network are neural networks. However, they have distinct parameters while sharing the same topology. The Q-estimated value of $Q(S, a)$ for the current state–action pair is generated by the computational network, which uses the most recent parameters. The Q-estimated value of $Q^+(S, a)$ is used to assess the DDQN loss function under the current channel condition–downlink scheduling mode. The target network does not update the parameters in real time, instead copying them from the computational network to the target network every specific iteration step c during the training time. Backpropagation and stochastic gradient descent (SGD) methods can be used to change the network parameters. DDQN loss function occurs under the current channel condition–downlink scheduling strategy. When the system is in the current channel uplink and downlink measurement state matrix S , the optimal state–action reward function $Q(S, a)$ in the downlink scheduling model, indicates the largest cumulative discount gain of completing scheduling action a' to enter the next state, S' . The revised phrase is written as follows:

$$Q(S, a) \leftarrow Q(S, a) + \delta [r(S, a) + \gamma Q(S', \max_{a'} Q(S', a')) - Q(S, a)] \quad (23)$$

where $\gamma = (0, 1) \in Q$ stands for the pace at which future incentives will diminish and $\delta = (0, 1) \in Q$ represents the learning rate. A computational network is utilized to imple-

ment the downlink adaptive scheduling procedure for NR-V2I communication once the network has been trained.

The DDQN-based downlink scheduling algorithm in this paper is shown in Algorithm 1:

Algorithm 1: Intelligent DDQN-based link scheduling algorithm for NR-V2I

Input: Calculate network weights θ ; target network weights $\hat{\theta} = \theta$.
Initialization: Memory database size N ;
 Step 1: Repeat the number of iterations $episode = 1$ to M do;
 Step 2: Initialize the state S_1 ;
 Step 3: for the number of subframes $t = 1$ to F do;
 Step 4: The action a_t that fulfills $a_t = \operatorname{argmax}_a Q(S_t, a; \theta)$ with probability ϵ , or the number of air division multiplexing layers r and the order D_{MCS} of MCS, is chosen by the E-IoV server;
 Step 5: E-IoV server schedules the corresponding number of layers r and the order of the MCS D_{MCS} for the downlink, and then calculates the reward value BER $B(S_t, a_t)$, and the system enters a new state $S_{t+1} = S'_t$;
 Step 6: The memory database stores the previous iteration experience $(S_t, a_t, B(S_t, a_t), S'_t)$;
 Step 7: Randomly select a small batch of sample data (S_t, a_t, B_t, S'_t) from the memory database and train the network; the target network obtains Q target value $Q^+(S, a)$, and the computational network obtains Q estimated value $Q(S, a)$;
 Step 8: If the final state is reached;
 Step 9: Then $Q^+(S, a) = r(S_t, a_t)$;
 Step 10: Otherwise, $Q^+(S, a) = r(S, a) + \gamma Q(S', \max_a Q(S', a'))$, γ is the decay rate of future rewards.
 Step 11: Calculate the loss function according to Equation (21) and update the weights of the computational network according to Equation (22);
 Step 12: Every certain number of iterations, update the parameters of the target network with the parameters of the computational network, setting $\hat{\theta}$ to $\hat{\theta} = \theta$;
 Step 13: end;
 Step 14: until the iteration termination condition is reached.
Output: DDQN downlink adaptive scheduling model.

3.3. Training Parameter Settings

The structure of the online learning and offline deployment phases of the DRL-based intelligent link scheduling method for NR-V2I cooperation is depicted in Figure 5.

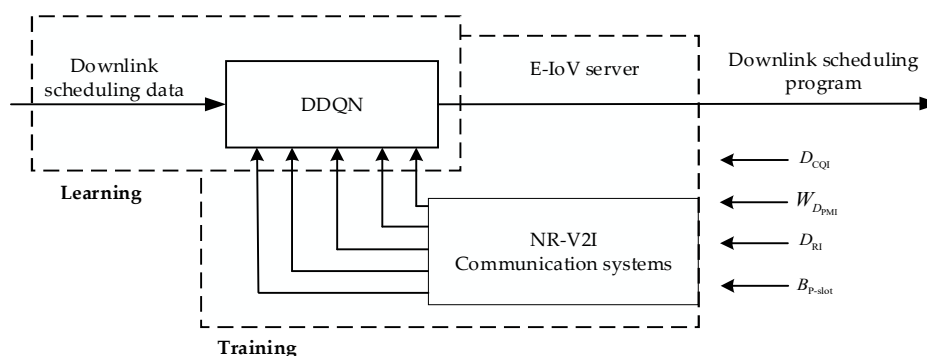


Figure 5. DQN-based reliable link scheduling framework.

Offline learning phase: The core of DDQN is training the neural network. To make the DDQN model applicable to various scenarios, sample downlink adaptive data from the NR-V2I communication system under various scenarios and parameters must be obtained. The DDQN model is then trained using these sample data.

This work considers two prominent cases—NR-V2I high-speed movement scenarios and scenarios with significant noise interference—where the performance of standard methods is more constrained for training and learning. Two different vehicle terminal

moving speeds are taken into consideration during the training process, and the data sets of these speeds are (60 km/h, 120 km/h), which used to train the DDQN downlink adaptive network for high-speed mobile scenarios. Different delay value data sets are also given consideration, with configured delays ranging from 0 to 15 with a step size of 1. The NR-V2I communication environment must be represented in an appearance that is consistent with the reinforcement learning environment in order to apply reinforcement learning techniques to the downlink adaptation challenge.

In this paper, the NR-V2I communication environment is constructed by using the matlab platform, and pytorch, an open-source deep learning framework, is employed to build and deploy the reinforcement learning component. The interaction between the data and the environment may be realized by using the python and matlab platforms. The training process can be described as a continuous interaction between the intelligent body and the environment for the intelligent body to choose the best course of action. An Intel(R) Xeon® E5-2678V3 CPU with 64 GB of RAM, an NVIDIA GeForce RTX2080Ti graphics card, and Python 3.9 and Pytorch 1.13 deep learning framework serve as the hardware and software platforms for the training. The training settings for the DQN system and the simulation parameters for the NR-V2I communication system are specified as indicated in Tables 1 and 2, respectively.

Table 1. Communication system parameter settings.

Parameter Name	Parameter Value
Carrier Frequency	5.925 GHz
Carrier Interval	30 kHz
Number of subcarriers	624
FFT Points	1024
Modulation mode	QPSK, 16 QAM, 64 QAM, 256 QAM
Channel model	TDL
Number of antennas of road test unit	32
Number of vehicle terminal antennas	4
Number of subframes	300

Table 2. DQN system training parameter settings.

Parameter Name	Parameter Value
Iteration number	1000
Memory size	1000
Frequency of update of target network parameters	150
Activation function	Tanh
Loss function	Huber
Learning rate	0.01
Batch size	16
Number of vehicle terminal antennas	0.9

The DNN is an input layer with σ nodes that are connected to the components of S ; there are five hidden layers with 64, 128, 256, 128, and 64 nodes, respectively; each hidden layer has a Tanh activation function; and there is an output layer with τ nodes. The structure is shown in Figure 6, where a_i denotes the value of the optimal downstream scheduling plan that the DNN has obtained. The last layer of the output adopts a fully connected layer, and the number of output nodes corresponds to the quantity of communication decisions given by the E-IoV server to the vehicle terminal.

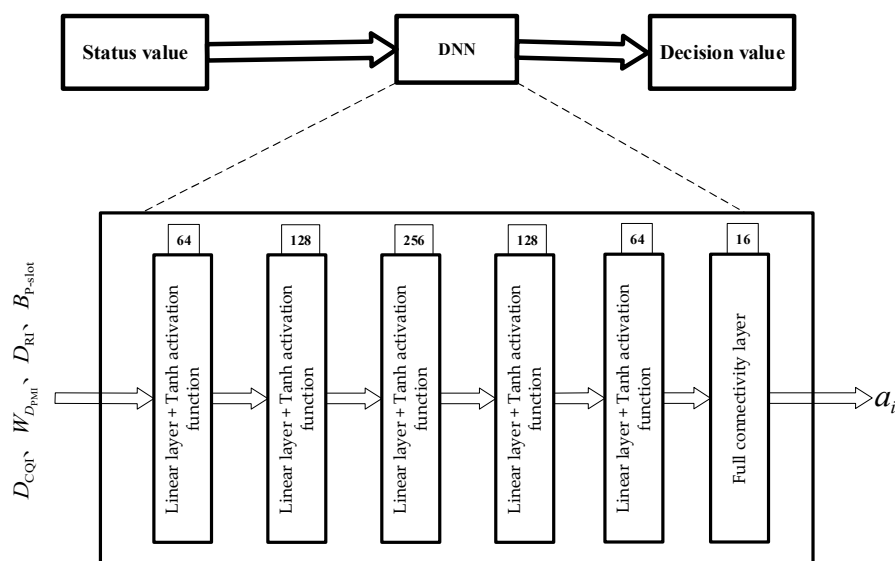


Figure 6. DNN layer network structure.

In this paper, the learning rate is specified in this study to be 0.01, and the future reward decay γ is specified to be 0.9. The modulation methods employed in the present investigation are QPSK, 16 QAM, 64 QAM, and 256 QAM. The channel model is the tapped delay line (TDL). The Adaptive Moment estimation (ADAM) technique, which can adaptively update the learning rate and SGD, can be employed to update the network parameters of the DQN network. The training of the network occurs when the sample data in the memory database reach 300 and continues until the network converges. A batch size of 16 indicates that 16 sample data are randomly selected from the memory database for training each time. The DQN network outputs the BER magnitude for all downlink transmission modes after network training is complete. The RSU then chooses the MCS and the number of air-division-multiplexing layers that, through the Q-learning principle, will yield the BER that is most suitable for downlink communication.

4. Simulation Results and Analysis

In this section, we compare the proposed algorithm to the OLLA, DQN, and NoOLLA algorithms in a typical high-speed moving scenario in order to assess how well the proposed algorithm performs in terms of average BER and throughput when used to schedule highly reliable intelligent downlinks in a 5G NR-V2I scenario. After simulating the algorithm using the primary communication system and DDQN network characteristics as described in Tables 1 and 2, Figures 7 and 8 display the simulation results for the average BER and throughput. Last, we compare the average number of iterations between DQN and DDQN.

In a 5G NR-V2I scenario, the vehicle often needs high data transmission reliability, particularly for security-related data transmissions, like traffic information and vehicle state updates. Because of the algorithm's low average BER performance, it may effectively lower the error rate of data transmission even when there is a high signal-to-noise ratio and a complex channel, increasing the dependability of data transmission. Signals may experience multiple path propagation in high-speed movement circumstances, leading to multipath effects. Signals can interpolate due to multipath effects, increasing the likelihood of intersymbol interference (ISI) and raising the BER. Different frequency components can result from high-speed movement due to selective fading of the signal at the frequency. This increases the BER of signal transmission and results in frequency-selective distortion. The BER performance of the methods at the same delay when the delay is in 0 or 10 μ s is shown in Figure 7a,b, respectively. The suggested method is 0.05, 0.07, and 0.1 lower than the average BER using DQN, OLLA, and NoOLLA, respectively, when the delay and frequency bias are 0 μ s and 436 Hz. The average BER performance of several algorithms under

doppler shifts of 250 Hz and 500 Hz, respectively, is shown in Figure 7c,d. In particular, the suggested DDQN method greatly improves the average BER performance at the same multispectral frequency shift. The suggested algorithm is 0.04, 0.08, and 0.1 lower than the average BER using the DQN algorithm, OLLA algorithm, and NoOLLA algorithm, respectively, when the frequency bias and time delay are 250 Hz and 9 μ s, respectively. Continuous action space issues can be handled with the OLLA algorithm. In order to avoid the complexity of directly searching for globally optimal actions, it separates the continuous action space into discrete local action spaces and employs local action selectors to choose actions. In contrast, using the continuous action space directly instead of the OLLA method typically entails spending more time and processing resources looking for global optimal actions. In order to develop better scheduling strategies in the high-dimensional state space and complex continuous action space of high-speed moving scenes, the OLLA algorithm can converge more quickly when compared to the NoOLLA algorithm.

The type of action space may affect how the OLLA and DQN algorithms affect the BER performance of communication link scheduling. The OLLA algorithm may be more appropriate if a continuous action space is involved because it can handle the problem of the continuous action space more effectively. However, due to the way the DQN algorithm handles the discrete action problem, it may be a superior fit for the discrete action space. Because the action space for the communication link scheduling problem is discrete, the DQN method may be a preferable choice for scheduling decisions because it performs better on average than the OLLA algorithm in terms of BER. The DQN algorithm is appropriate for the discrete action space problem because it uses knowledge of the Q value function to choose actions that can reduce average BER. The DDQN algorithm is an enhancement to the DQN method that may select the action strategy in the situation of discrete action space more correctly, thereby lowering the average BER even more.

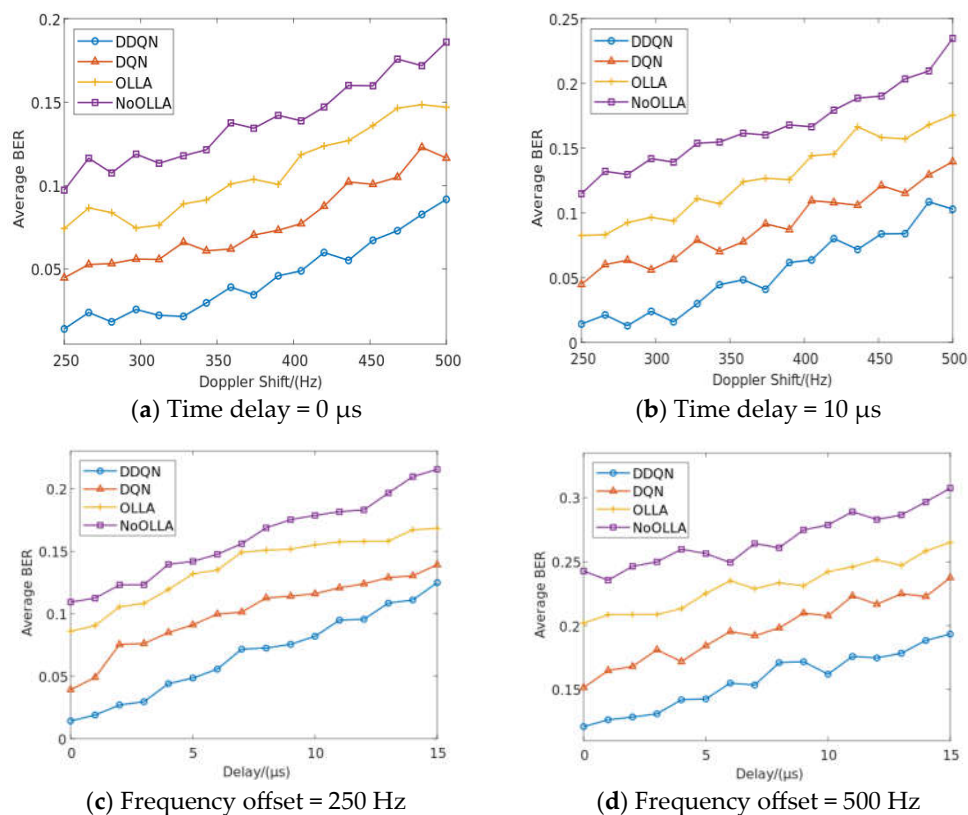


Figure 7. Average BER performance of different algorithms in high-speed moving scenarios.

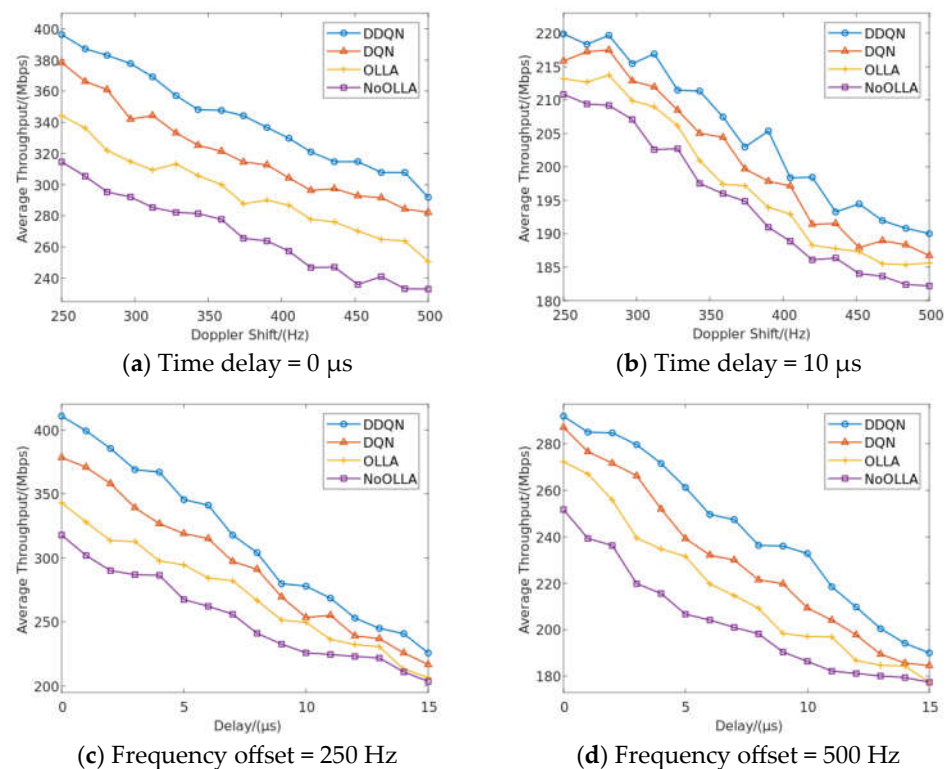


Figure 8. Throughput performance of different algorithms in high-speed mobility scenarios.

A highly efficient scheduling algorithm can optimize resource allocation, increase anti-interference performance, increase resource utilization, and adapt to dynamic environments, resulting in a significant increase in communication link throughput. A system with high throughput can process data transmission more quickly and boost the network's data transmission effectiveness. The algorithm's average BER performance benefits in 5G NR-V2I scenarios are primarily seen in the high dependability, potent anti-interference, self-adaptability, and high throughput it offers. These benefits will improve the efficiency and reliability of 5G vehicle communication, enabling stable and reliable data transmission between vehicles and infrastructure in a challenging wireless communication environment. Figure 8a,b depict the throughput performance of the various methods for delays of 0 μs and 10 μs, respectively, and the effectiveness of the suggested DDQN algorithm at a certain delay. The suggested algorithm is 22 Mbps, 61 Mbps, and 88 Mbps higher than the throughput of the DQN algorithm, OLLA algorithm, and NoOLLA algorithm, respectively, when the delay and frequency bias are 0 μs and 281 Hz. The throughput performance of several methods under doppler shifts of 250 Hz and 500 Hz is shown in Figure 8c,d, respectively. Among these, the suggested DDQN algorithm's throughput performance at the same multispectral shift is much enhanced. The throughput using the suggested method is 26 Mbps, 51 Mbps, and 78 Mbps higher than the throughput using the DQN algorithm, OLLA algorithm, and NoOLLA algorithm, respectively, when the frequency bias and time delay are 250 Hz and 0 μs, respectively. The OLLA algorithm has the flexibility to optimize local action selection under dynamic channel and network conditions, improve resource consumption efficiency, and increase throughput. If the OLLA algorithm is not used when scheduling the communication connection or if the search in the continuous action space or discrete action space is not efficient or flexible enough, the throughput of the link may be impacted. For the discrete action space problem, the DQN algorithm works better. It is better suited for highly reliable intelligent downlink scheduling in 5G NR-V2I scenarios by learning the Q value function to choose the actions that can maximize throughput. The DDQN algorithm used in this research may better optimize the link

resource allocation and increase the throughput of communication lines by lowering the overestimation of the Q value.

Finally, the average iterations of DQN and DDQN are compared. Comparing the average number of iterations helps identify which algorithms converge faster to a suitable performance level under the same training conditions. Fewer iterations usually indicate a more efficient training process. In addition, fewer iterations may mean that the training process is more stable, which also means that the algorithm requires fewer computational resources. As shown in Table 3, although DQN is less than DDQN in the number of iterations, DDQN is more stable when the environment deteriorates, because its number of iterations changes more slowly.

Table 3. System training duration.

Parameter Name	Performance Name	DQN Average Iterations	DDQN Average Iterations
Time delay = 0 μ s	Average BER	732	761
Time delay = 10 μ s	Average BER	775	784
Frequency offset = 250 Hz	Average BER	753	789
Frequency offset = 500 Hz	Average BER	794	810
Time delay = 0 μ s	Average Throughput/(Mbps)	703	732
Time delay = 10 μ s	Average Throughput/(Mbps)	731	747
Frequency offset = 250 Hz	Average Throughput/(Mbps)	726	752
Frequency offset = 500 Hz	Average Throughput/(Mbps)	765	773

5. Conclusions

This article suggests an ultra-reliable intelligent downlink scheduling technique based on DDQN for the 5G NR-V2I autonomous driving scenario. With D_{CQI} , D_{RI} , and D_{PMI} from the measurement feedback of the vehicle terminal and the statistics B_{P-slot} as input variables, this approach combines the DNN network and Q-learning algorithm. The BER for all downstream transmission modalities is the output. According to the Q-learning concept, the RSU chooses the MCS and the number of multiplexing layers with the lowest BER for downlink transmission. In order to avoid imperfection in the learning process or noise in the data that may lead to bias, this paper uses appropriate data preprocessing methods to reduce the impact of noise, such as filtering or smoothing. In this paper, the empirical replay mechanism is used to reduce the problem of high Q overestimation. In order to reduce the cost of two independent networks, this paper adopts some techniques to reduce the training cost, such as sharing some parameters and reducing the network size. In order to avoid a DDQN that may lead to over-exploitation and less exploration, this paper uses appropriate exploration strategies, such as the ϵ -greedy strategy, to ensure that the algorithm maintains a certain degree of exploration. In order to avoid policy oscillations that may be caused by managing two Q networks, this paper uses a soft update or progressive update to smooth the policy update process. In order to avoid overfitting problems, this paper uses techniques, such as regularization and stopping training in advance, to avoid overfitting.

The simulation demonstrates that the ultra-reliable intelligent downlink scheduling algorithm based on DDQN outperforms the NoOLLA, OLLA, and DQN algorithms in terms of average error rate and throughput performance, ensuring the ultra-reliability and efficiency of communication between vehicles and infrastructure. In addition, although DQN is less than DDQN in the number of iterations, DDQN is more stable when the environment deteriorates, and its number of iterations changes more slowly. In future research, we will consider the use of appropriate state representation methods by using recurrent neural network (RNN) or other timing models to deal with dynamic environments to cope with the training difficulties that may be caused by highly dynamic environments. Considering that the algorithm update under the condition of real-time change may require a lot of computing resources, the use of distributed computing can be considered to improve computing efficiency. In order to ensure the stability of the system quickly adapted to new

conditions, a buffer zone or sliding window can be considered to slow down the adaptation speed of the model to maintain the stability of the system.

Author Contributions: Conceptualization, Y.L. (Yong Liao); Methodology, J.W. (Jizhe Wang) and Y.L. (Yong Liao); Software, Y.J., Z.S. and Y.Z.; Validation, Y.J. and Y.L. (Yu Luo); Formal analysis, Y.Z.; Investigation, J.W. (Jian Wang); Data curation, Z.S. and L.T.; Writing—original draft, J.W. (Jizhe Wang), J.W. (Jian Wang), Y.J. and Y.L. (Yu Luo); Writing—review and editing, Y.J. and Y.L. (Yu Luo); Visualization, Y.J.; Supervision, J.W. (Jizhe Wang) and Y.L. (Yong Liao); J.W. (Jizhe Wang), literature research and introduction writing, overall paper framework design, overall guidance for this paper; Y.Z., establishment and analysis of DDQN mathematical model; J.W. (Jian Wang), drawing and description of 5G NR-V2I scene diagram and communication system diagram; Z.S., simulation system code implementation and result analysis; L.T., DDQN-based downlink scheduling for 5G NR-V2I pseudocode writing and simulation parameter settings. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to privacy, we can not provide the data.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, C.; Yao, G.; Liu, L.; Pei, Q.; Song, H.; Dustdar, S. A Cooperative Vehicle-Infrastructure System for Road Hazards Detection with Edge Intelligence. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5186–5198. [CrossRef]
- Chen, C.; Wang, C.; Li, C.; Xiao, M.; Pei, Q. A V2V Emergent Message Dissemination Scheme for 6G-Oriented Vehicular Networks. *Chin. J. Electron.* **2023**, *32*, 1179–1191.
- Yan, X.; Liu, G.; Wu, H.-C.; Zhang, G.; Wang, Q.; Wu, Y. Robust Modulation Classification Over α -Stable Noise Using Graph-Based Fractional Lower-Order Cyclic Spectrum Analysis. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2836–2849. [CrossRef]
- Mota, M.P.; Araujo, D.C.; Costa Neto, F.H.; de Almeida, A.L.F.; Cavalcanti, F.R. Adaptive Modulation and Coding Based on Reinforcement Learning for 5G Networks. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
- Tsurumi, S.; Fujii, T. Reliable Vehicle-to-Vehicle Communication Using Spectrum Environment Map. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 310–315.
- Huang, Z.; Zheng, B.; Zhang, R. Roadside IRS-Aided Vehicular Communication: Efficient Channel Estimation and Low-Complexity Beamforming Design. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5976–5989. [CrossRef]
- Veres, M.; Moussa, M. Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3152–3168. [CrossRef]
- Ramjee, S.; Ju, S.; Yang, D.; Liu, X.; Gamal, A.E.; Eldar, Y.C. Fast Deep Learning for Automatic Modulation Classification. Available online: <https://arxiv.org/abs/1901.05850v1> (accessed on 3 July 2023).
- Liu, X.; Yang, D.; Gamal, A.E. Deep Neural Network Architectures for Modulation Classification. In Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 915–919.
- Klaine, P.V.; Imran, M.A.; Onireti, O.; Souza, R.D. A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2392–2431. [CrossRef]
- de Carvalho, P.H.P.; Vieira, R.D.; Leite, J.P. A Continuous-State Reinforcement Learning Strategy for Link Adaptation in OFDM Wireless Systems. *J. Commun. Inf. Syst.* **2015**, *30*, 47–57. [CrossRef]
- Robust Adaptive Modulation and Coding (AMC) Selection in LTE Systems Using Reinforcement Learning. Available online: <https://ieeexplore.ieee.org/abstract/document/6966162/> (accessed on 3 July 2023).
- Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
- Zhang, L.; Tan, J.; Liang, Y.-C.; Feng, G.; Niyato, D. Deep Reinforcement Learning-Based Modulation and Coding Scheme Selection in Cognitive Heterogeneous Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3281–3294. [CrossRef]
- Liao, Y.; Yang, Z.; Yin, Z.; Shen, X. DQN-Based Adaptive MCS and SDM for 5G Massive MIMO-OFDM Downlink. *IEEE Commun. Lett.* **2023**, *27*, 185–189. [CrossRef]
- Xi, L.; Yu, L.; Xu, Y.; Wang, S.; Chen, X. A Novel Multi-Agent DDQN-AD Method-Based Distributed Strategy for Automatic Generation Control of Integrated Energy Systems. *IEEE Trans. Sustain. Energy* **2020**, *11*, 2417–2426. [CrossRef]
- Bui, V.-H.; Hussain, A.; Kim, H.-M. Double Deep Q-Learning-Based Distributed Operation of Battery Energy Storage System Considering Uncertainties. *IEEE Trans. Smart Grid.* **2020**, *11*, 457–469. [CrossRef]

18. Xiaoqin, S.; Juanjuan, M.; Lei, L.; Tianchen, Z. Maximum-Throughput Sidelink Resource Allocation for NR-V2X Networks with the Energy-Efficient CSI Transmission. *IEEE Access* **2020**, *3*, 73164–73172. [[CrossRef](#)]
19. Kim, J.; Choi, Y.; Noh, G.; Chung, H. On the Feasibility of Remote Driving Applications Over mmWave 5G Vehicular Communications: Implementation and Demonstration. *IEEE Trans. Veh. Technol.* **2023**, *9*, 2009–2023. [[CrossRef](#)]
20. Liao, Y.; Yin, Z.; Yang, Z.; Shen, X. DQL-based intelligent scheduling algorithm for automatic driving in massive MIMO V2I scenarios. *China Commun.* **2023**, *3*, 18–26. [[CrossRef](#)]
21. Sasaoka, N.; Sasaki, T.; Itoh, Y. PMI/RI Selection Based on Channel Capacity Increment Ratio. In Proceedings of the 2019 International Symposium on Multimedia and Communication Technology (ISMTC), Quezon City, Philippines, 19–21 August 2019; pp. 1–4.
22. Passive Method for Estimating Available Throughput for Autonomous Off-Peak Data Transfer. Available online: <https://www.hindawi.com/journals/wcmc/2020/3502394/> (accessed on 19 July 2023).
23. Zhao, D.; Qin, H.; Song, B.; Zhang, Y.; Du, X.; Guizani, M. A Reinforcement Learning Method for Joint Mode Selection and Power Adaptation in the V2V Communication Network in 5G. *IEEE Trans. Cogn. Commun.* **2020**, *3*, 452–463. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.