CURIOSITY DRIVEN RESOURCE ALLOCATION FOR 5G AND BEYOND VEHICULAR NETWORKS

by

Baorui Jia

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
March 2024

To my grandfather "Jia Rongfang."
For all the love and support you have provided.
Until we meet again.

Contents

List of Tables

List of Figures

Abstract

With the rapid advancement of the Internet of Vehicles (IoV), there arises an increasing demand for efficient connectivity and communication mechanisms between vehicles and infrastructures, wherein resource allocation assumes paramount importance. The primary objective of a resource allocation algorithm is to distribute limited resources, including power and spectrum, to mobile devices within the network while catering to the diverse requirements of users. In this thesis, we introduce a novel approach called the Intrinsic Curiosity Module (ICM) based Double Q Learning (DQL) for resource allocation, denoted as ICM-DQRA, aimed at addressing resource allocation challenges in IoV network. We integrate the ICM into the DQL algorithm to incorporate an intrinsic reward to the agent. This intrinsic reward, absent in most reinforcement learning algorithms, serves to incentivize the agent to explore the environment further and make decisions conducive to better rewards. Through comprehensive simulations, our proposed method outperforms other approaches, such as the greedy method and DQL method. Specifically, the ICM-DQRA algorithm achieves a more efficient resource allocation among vehicles, leading to a substantial reduction in energy consumption across the network, ranging from 20% to 27%.

Acknowledgements

Thank you Dr. Yujie Tang, for your time and your guidance. Thank you to my readers, Dr. Qiang Ye and Dr. Samer Lahoud, for your careful reading and your helpful comments. Thank you Dr. Corey DeGagne, for offering me the opportunity to learn from and work with you as a teaching assistant in your courses. Thank you Dr. Fangda Cui, for pushing me to do more when I was willing to settle for less. Thank you to someone who devoted a great deal of their time and energy to me.

Chapter 1

Introduction

1.1 Background

Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications are key parts in the context of Internet of Vehicles (IoV), which falls under the umbrella of Internet of Things (IoT). Due to the rapid development of communication technologies and the IoT, there has been an increase in the need for network resources to accommodate for efficient and reliable communication among vehicles for various purposes such as safety, traffic management, and navigation services. Therefore, it is important to be able to effectively manage the limited resources available for communication. Current V2V resource allocation research primarily focuses on optimizing the allocation of resources such as bandwidth, power, and spectrum in vehicular communication networks [24]. And this field of research has gained increasing attention around the world with the emergence of connected and autonomous vehicles.

IoV represents a paradigm where vehicles are interconnected with each other and with infrastructure elements to enable advanced communication and collaboration. V2V communication allows vehicles to exchange information directly with neighboring vehicles, facilitating cooperative driving, collision avoidance, and traffic controls. Real-time data exchange is a key component to making informed decisions and enhancing overall road safety and efficiency. In addition to V2V communication, IoV also relies on V2I communication, which involves interactions between vehicles and roadside infrastructure such as traffic lights, road signs, and base stations, etc. V2I communication enables vehicles to access information about traffic patterns and infrastructure updates, thus enabling enhanced traffic management. As IoV technologies continue to evolve, the integration of V2V and V2I communication can greatly improve transportation systems, making them safer, more efficient, and more environmentally friendly.

One bottleneck for the fifth-generation (5G) cellular technology and beyond research is the high energy consumption caused by a large number of devices with high transmission rates [10]. In response to the challenge, various reinforcement learning (RL) based algorithms were proposed.

1.2 Research Objective

The objective of this thesis is to develop a novel resource allocation scheme that better suits the needs of 5G networks. Specifically, we propose to use an Intrinsic Curiosity Module (ICM) and build it onto double Q-learning algorithm, to create a new RL-based resource allocation mechanism. This ICM agent consists of three primary components: a feature extractor model, a forward model, and an inverse model.

In the feature extractor module, we utilize a Recurrent Neural Network (RNN) model to obtain a feature representation for the states of my agent. The motivation for this feature extractor is, we observed that the states of the agent, though very descriptive and relevant to the agent itself, have the tendency to be identical between two consecutive time steps. One reason is that to better simulate 5G scenario, we set the time step to be 0.01 second (10 milliseconds), which can be too short for channel information to change. This is one aspect that gives rise to an exploration challenge to our agent. The goal of this RNN-based feature extractor is to obtain a feature representation of the states that can better motivate the agent to explore the environment.

In the forward model, we employ a Multilayer Perceptron (MLP) based architecture to predict the feature representation of next state, and calculate the intrinsic reward based on the difference between this predicted and the actual feature representation from the feature extractor model.

The inverse model is also MLP-based, and it provides a way to train the ICM agent. Specifically, this model predicts the action the agent would take with feature representations as input, and the objective for training is to minimize the Mean Squared Error (MSE) between this predicted action and the actual action the agent takes. As long as this MSE error is small enough, we are able to tell that the ICM agent learned a meaningful feature representation of the states.

To conclude, we propose a new strategy for V2V resource allocation with the goal of reducing the energy consumption for the entire vehicular network. This proposed strategy involves the application of an ICM agent. The ICM agent motivates the agent to explore the environment through an intrinsic reward so that the agent would take actions it wouldn't have taken without it. The new actions the agent takes has the potential to lead to better results.

Firstly, we use Double-Q learning method in 4G and 5G settings to solve the resource allocation problem and compare their performance. Additionally, we apply deep Q learning algorithm as a baseline performance. Following that, we sample data from the memory buffer to train the ICM agent. With the trained ICM agent, we are able to calculate the intrinsic reward for any action the agent takes, and this intrinsic reward is usually absent in most reinforcement learning algorithms. In the next phase, we apply ICM-DQRA method that takes into account the intrinsic reward of the agent, and train the new network with all other parameters stay the same under 5G setting. Finally, we evaluate the performance of the proposed ICM-DQRA resource allocation scheme and compare its performance with existing solutions under identical simulation scenarios.

The main contributions of this thesis can be summarized as follows:

- We propose ICM-DQRA, an ICM-based resource allocation scheme to optimize the energy consumption in a vehicular network while satisfying the latency constraints. Our approach incorporates an intrinsic reward, which is usually absent in most reinforcement learning algorithms.

- We further tailor the architecture of the ICM agent for our problem. It can be treated as a built-on module to the double Q-learning algorithm and have the flexibility to fit into different simulation settings. This means that we do not need to re-train the agent when we change parameters such as number of vehicles during simulation.

- The proposed ICM-DQRA algorithm motivates the agent to explore a larger action space and after comprehensive evaluations, we observe that this mechanism gives a less aggressive power selection result, and significantly reduce the energy consumption for the entire network while satisfying the latency constraints at

the same time.

1.3 Thesis Outline

The rest of the thesis is organized in the following manner:

In chapter two, we present a brief overview on resource allocation problems and argue that reinforcement learning based algorithms are by far one of the most promising solutions to modern resource allocation problems. Then we give a brief overview of reinforcement learning, including its components and categories. We also compare the major difference between the fourth-generation (4G) research versus 5G and beyond research.

Chapter three introduces the details of ICM-DQRA learning approach for V2V communications, including the motivation for the Intrinsic Curiosity Module (ICM), the fundamental framework, and the detail of the proposed algorithm. In addition, this chapter formally formulate the problem we are solving.

Chapter four first provides detailed information on the training and testing procedures of the proposed algorithm, along with the evaluation metrics. After the necessary information is provided, the simulation results for ICM-DQRA are provided with comparison and analysis.

In the last chapter, the thesis is concluded with current achievements, and potential future research directions are discussed.

Chapter 2

Related Work

In this chapter, we first discuss selected work related to resource allocation in vehicular networks. Then we introduce the background knowledge for Reinforcement Learning (RL) as well as Deep Reinforcement Learning (DRL) and analyze why these methods become a promising solution to resource allocation problems. Lastly, we compare the major difference between the fourth generation (4G) cellular technology and 5G beyond research. The research work done in this thesis focuses primarily on 5G and beyond research, but given the fact that there are still many mobile devices around the world that use the 4G Long Term Evolution (LTE) technologies, we include 4G scenario for comparison purposes.

2.1 Resource Allocation

In recent years, there has been a growing body of research dedicated to addressing resource allocation problems aimed at enabling reliable, efficient, and safe communication among vehicles and infrastructures. The objectives of research can be broadly categorized into four categories: efficiency optimization [26], safety enhancement [12], traffic management [3], and cooperative driving [17] [22]. Each of these aspects focuses on specific challenges in resource allocation for vehicular networks, but share the same goal to satisfy the Quality of Service (QoS) requirements.

Efficiency optimization strives to maximize the utilization of available resources in the vehicular network. The majority of existing algorithms and technologies in this field are geared towards maximizing throughput, minimizing packet loss, or reducing energy consumption. The ultimate goal is to improve network efficiency that would lead to better overall performance. Additionally, these strategies also facilitate the integration of advanced applications and services that adds to the effectiveness of modern transportation systems.

Safety enhancement aims to prioritize communication for safety-critical applications such as navigation services, alarms, etc. to facilitate collision avoidance. By allocating resources dynamically based on the time sensitiveness of messages and the availability of spectrum resources, the likelihood of traffic accidents can be reduced. Numerous algorithms, along with their variations, have been proposed to cater to the requirements of high reliability and low latency communication in vehicular networks. These solutions aim to ensure that time-sensitive alarm information can be promptly broadcast to incoming vehicles.

Traffic management centers around facilitating real-time traffic monitoring and congestion control by allocating resources intelligently. This aspect involves optimizing the use of road infrastructure. By leveraging data from sensors, cameras, and connected vehicles, traffic management authorities can gain insights into traffic patterns, so as to adjust traffic signals timings to reroute traffic. This allows for more efficient resource allocation. Furthermore, it not only reduces congestion but also enhances safety, and contributes to the general improvements in urban mobility.

Cooperative driving focuses on facilitating the exchange of vital information such as speed, position, and intentions among vehicles, allowing them to make collective decisions. This aspect involves many applications such as coordinated merging that allows for vehicles merging into a traffic flow without disrupting it, and intersection management where vehicles pass through intersections safely and efficiently with information sharing. With the support of V2V and V2I communications, cooperative driving is able to enable low latency in the transmission of messages and contribute to autonomous driving systems in the future.

Traditional resource allocation algorithms include greedy algorithms [2], priority scheduling algorithms [19], etc. They were widely used in resource allocation problems due to their simplicity and ease of implementation. However, they have several limitations. For example, greedy algorithms is likely to reach an local optimal choice at each step and converge to a sub-optimal solution. This locally optimal solution is unlikely to be the best global solution for a given problem. For priority scheduling algorithms, their complexity is likely to become unmanageable as the number of tasks or priority levels increase. This could lead to overhead in terms of system resources

and processing time, particularly in a dynamic vehicular environment where priority levels are adjusted frequently. In fact, most traditional algorithms would suffer from the dynamic environments and the algorithms themselves would need frequent adjustment to remain effective in such scenarios.

With the development of communication technologies, researchers have been actively seeking for modern algorithms that can handle the challenges in resource allocations problems. RL-based algorithms [11] have been proposed and proved to have promising results.

2.2 Reinforcement Learning

Reinforcement learning is a sub-field of machine learning. In each iteration, the agent learns to make decisions by interacting with the environment. The agent takes actions based on its current state and receives feedback in the form of rewards or penalties from the environment [4]. This iterative process goes on with the goal to learn a policy that maximizes cumulative reward over time. Though it might happen that in certain cases, the environment is fully known, such as a well-defined grid environment, and it becomes unnecessary for the agent to explore the environment any more. But in most real-world reinforcement learning environments, this is unrealistic.

Key components in reinforcement learning [4] include:

- Agent: The agent would interact with the environment in an reinforcement learning algorithm, and is usually varies in different scenarios. It also depends on the problem formulation that we have. In resource allocation problems for IoV, one common practice is to treat a V2V link as an agent [25]. We adopt this tradition in this thesis. The agent is usually equipped with decision-making capabilities and can learn to take actions based on its rewards from previous interactions.

- Environment: The environment represents the external domain in which the agent learns and takes actions. It can also vary widely, and in resource allocation problems, one common practice is to treat everything beyond the particular V2V link as the environment [25]. This environment contains collective information that is related to the agent's decision making.

- State: The state represents the current configuration of the environment at a specific point in time. It captures all relevant information that the agent needs to make decisions, and in resource allocation problems, one common practice is to concatenate information such as channel information, interference, selected sub-channels from previous time slots into a unified state representation [25]. Additionally, states can be discrete or continuous, depending on the nature of the problem domain. The states considered in this thesis are continuous.

- Action: The action is a decision made by the agent within a given state of the environment, and it directly impacts the environment. The action space contains the set of possible choices available to the agent. Actions can also be discrete or continuous that represents any form of the agent's interaction with the environment. The actions in this thesis are considered to be discrete.

- Reward: Reward serves as feedback from the environment to the agent and guides the learning process by reinforcing or discouraging certain behaviors. The agent's goal is to maximize its cumulative reward over time by taking actions based on its learned policy. Rewards can also be in the form of penalties, which would discourage the agent from taking certain actions. The reward in this thesis is jointly determined by the capacities of V2V and V2I links as well as the latency constraint of the considered V2V link.

Generally, we can categorize RL algorithms into two broad types: model based and model free algorithms [9], as shown in figure 2.1. The agents learn an explicit model of the environment's dynamics and take actions based on it in a model based algorithm, whereas in model free algorithms, the agents learn directly through interactions with the environment without the need to model its dynamics. The model free algorithms can be further divided as value based methods and policy based methods.

The value based methods aim to estimate the value of state-action pairs and select actions based on these estimates. Examples include Q-learning, Deep Q-Learning (DQL), and its variants. In Q-learning, the Q-values are updated using the Bellman equation:

$$Q(s,a) \leftarrow (1-\alpha) \cdot Q(s,a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s',a')) \tag{2.1}$$

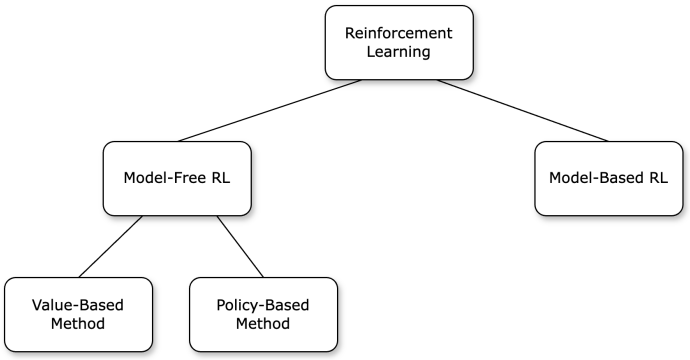

Figure 2.1: Categories of Reinforcement Learning Algorithms

that combines the immediate reward with discounted maximum Q-value for the subsequent state [8]. Here $Q(s, a)$ is the Q-value for state-action pair (s, a), α is the learning rate between 0 and 1, r is the immediate reward after taking action a, γ is the discount factor and $Q(s', a')$ represents the estimate of Q-value for next state-action pair. This approach has become the basis of many reinforcement learning algorithms.

However, this method faces challenges in that the Q-table for rewards will grow considerably large in a multi-agent environment with two or more agents [13], therefore it would require a large memory space for storage. This makes effective learning challenging. To handle this, Google Deep Mind developed deep Q-learning, which combines Convolution Neural Networks (CNN) with Q-learning that instead of expressing the value function for each state, it employs an approximation function using CNN [7]. For an n-dimensional state space and an action space with m possible actions, the neural network is essentially a function from R^n to R^m. Two main parts of a DQN network are experience replay that randomizes over the data and an iterative update that adjusts the action-values (Q) towards target values [15], which is a second Q-network called target Q-network. The target used by the Q-network and target Q-network are defined by equation (2.2) and equation (2.3) respectively:

$$Y_t^Q = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta_t) \tag{2.2}$$

$$Y_t^{DQN} = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta_t^-) \tag{2.3}$$

Here R_{t+1} represents the reward obtained after taking action a in state S_t, θ and θ^- represent the parameters for the main network and the target network, $Q(S_{t+1}, a; \theta_t)$

and $Q(S_{t+1}, a; \theta_t^-)$ represent the estimated Q value for the next state S_{t+1} with all possible actions a in the main and target networks respectively, we take action that leads to the maximum value.

However, one problem that might occur is during action selection, the deep Q-learning algorithm chooses the action with the highest Q-value. Whereas during action evaluation, it uses the same Q-value to estimate the long-term return, this makes it more likely to select overestimated values, resulting in overoptimistic value estimates. To avoid such problem, double Q-learning [23] is proposed to decouple the selection from the evaluation. With double Q-learning, we can rewrite the targets in equations (2.2) and (2.3) as:

$$Y_t^Q = R_{t+1} + \gamma Q(S_{t+1}, \max_a Q(S_{t+1}, a; \theta_t); \theta_t) \tag{2.4}$$

and

$$Y_t^{\text{DoubleQ}} = R_{t+1} + \gamma Q(S_{t+1}, \max_a Q(S_{t+1}, a; \theta_t); \theta_t') \tag{2.5}$$

respectively. This indicates that in double Q-learning, although we still use the main network's parameters θ_t to select actions, we use a second set of parameters θ_t' to fairly evaluate the value of the selected action. This second set of weights can be updated symmetrically by switching the roles of θ_t and θ_t' [23]. This will give us the following update rules for the Q-functions:

$$Q(s, a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha(s, a) \left(r + \gamma Q'(s', a')\right) \tag{2.6}$$

$$Q'(s, a) \leftarrow (1 - \alpha) \cdot Q'(s, a) + \alpha(s, a) \left(r + \gamma Q(s', a')\right) \tag{2.7}$$

Here $Q(s, a)$ represent the Q–function for the main networks and $Q'(s, a)$ represent the Q-function for the target network. Each Q-function is updated with the value of another Q-function, and the update is performed alternatively between them. This approach helps mitigate the overestimation bias in deep Q-learning.

The policy based methods directly parameterize the policy, which is a mapping from states to actions, and optimize it to maximize expected rewards. Examples include Actor-Critic methods and its variants.

Another topic related to RL that worth mentioning is the exploration-exploitation dilemma. Exploration refers to the agent's strategy of trying out different actions to

discover potentially better outcomes. Whereas exploitation refers to the agent's strategy of selecting actions that are known to yield high rewards based on past experience. Balancing exploration and exploitation is crucial for effective learning. There are several attempts that aim to address such dilemma in resource allocation problems, for example, Ding et al. proposed the attention methods (AMARL) in [5] that satisfies the requirements of a high rate for V2I links and low latency for V2V links. The proposed AMARL-based approach also has an excellent adaptability to environmental change. Apart from that, Shi et al. [21] proposed a sparse code multiple access-based centralized resource allocation scheme to address the challenge of high-speed vehicles across the coverage regions of multiple cells in 5G systems. These existing methods focus primarily on helping the agent learn an exploration strategy that is robust to the always changing environment. However, the cause of an exploration-exploitation dilemma varies and the changing environment is just one among many. In this thesis, we incorporate ICM with DQL that incentives the agent to explore a larger action apace that can lead to better rewards. This is achieved by providing an intrinsic reward to the agent, which is absent in most RL algorithms.

2.3 From 4G to 5G and Beyond

Both 4G and 5G networks are wireless communication technologies are used for mobile telecommunications. They support data transmission, enable users to access the internet, stream multimedia content, make voice calls, and use various applications on their mobile devices. Additionally, 5G networks are designed to be backward compatible with 4G networks, facilitating smoother transitions and coexistence with existing 4G networks during the deployment phase.

They also differ in aspects such as throughput, latency, and more. 5G networks promise significantly faster data rates compared to those offered by 4G networks. Moreover, 5G networks aim to achieve ultra-low latency, reducing the delay between sending and receiving data packets. This is crucial for real-time applications such as virtual reality, autonomous vehicles. Additionally, 4G networks primarily operate in lower frequency bands (sub-6 GHz), 5G networks utilize a wider range of frequency bands, including sub-6 GHz and millimeter-wave (mmWave) bands to achieve higher data rates and capacity.

Another important difference between 4G and 5G lies in the physical layer where dynamic power level assignment are supported [27]. This refers to the process of dynamically adjusting the transmit power levels of communication devices. By continuously monitoring channel conditions, neighboring cell activity, and user traffic patterns, this technique provides advantages with respect to minimizing interference and maximizing signal-to-interference-plus-noise ratio (SINR), and enhancing the overall quality of services (QoS). However, considering the compatibility between 5G and 4G networks, we adopt discrete power level assignment technique to simplify network planning and optimization process, but the general trend is expected to be the same for continuous power adaptation, which we aim to double check as a future work.

Currently, 5G networks are still being deployed with focus on deployments for urban areas and densely populated regions. At the same time, 4G networks are still widely used in many regions worldwide. Although 5G networks offer significant benefits, there are challenges associated with their deployment, including infrastructure costs, spectrum allocation, and energy consumption, etc.

Research on 5G and beyond is concentrated on several key areas to meet the growing demand for energy-efficient wireless networks and devices. One of these areas is dynamic spectrum management [16]. This research area aims to optimize spectrum utilization and minimize energy consumption. Strategies include spectrum sharing strategies, dynamic spectrum access policies, etc. to efficiently allocate spectrum resources based on real-time demand and environmental conditions.

Chapter 3

Learning with Curiosity

A Markov Decision Process (MDP) offers a formal approach to modeling decision-making problems. It relies on the Markov property, which stipulates that the future state depends solely on the current state and action, without consideration for the entire history of previous states and actions. It is common practice to formulate reinforcement learning algorithms as a MDP as the goal of a RL algorithm is for the agent to learn optimal policies through interactions with the environment [9].

In this chapter, we delineate our methodology for resource allocation in vehicular networks. We first give a brief overview of selected resource allocation schemes. Then, we formulate the resource allocation problem as MDP. After that, we present the detailed structure of the proposed ICM-DQRA framework, including the network architecture, the workflow of the designed algorithm, as well as the training and testing algorithms for the proposed methodology.

3.1 Schemes for Resource Allocation

The workflow typically involves the agent observing current state, taking certain action based on this observation, and receiving feedback in the form of rewards or penalties. Through iterative learning, the agent adjusts its behaviour, and in our case, resource allocation strategy, to maximize its long-term objectives. This process enables the agent to adapt to improve its resource allocation decisions over time and eventually "learns" to allocate resource intelligently. Figure 3.1 shows the general structure of reinforcement learning algorithms.

To handle the complexity and dynamics in modern resource allocation problems, reinforcement learning based methods have been adopted by researchers for its ability to learn from experience, coupled with its adaptability to varying environments, where traditional optimization techniques may fall short. We present the structure of selected resource allocation schemes in this section.

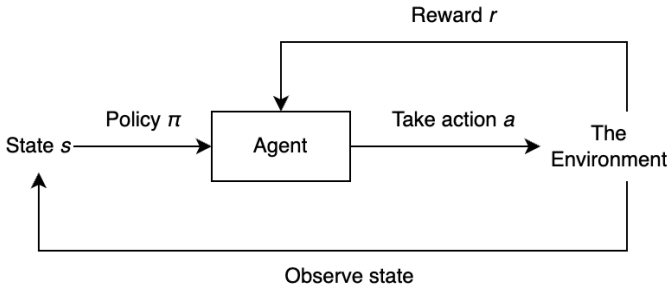

Figure 3.1: General Structure of Reinforcement Learning Algorithms

3.1.1 Q Learning

Q learning is a value-based model-free algorithm that learns to make optimal decisions in an environment by iteratively updating its Q-table. The Q-table stores the cumulative rewards for each state-action pair. During training process, Q-learning algorithm gradually converges to an optimal policy that maximizes cumulative rewards over time. This approach has become the basis of many reinforcement learning algorithms. Figure 3.2 shows the structure of Q-learning algorithm.

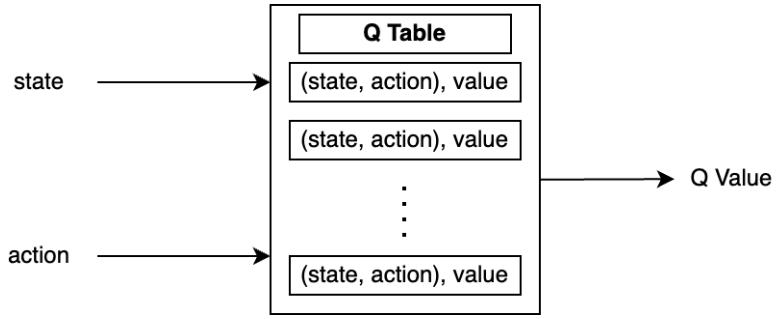

Figure 3.2: Q-learning Workflow

3.1.2 Deep Q Learning

Deep Q-learning (DQN) is also a value-based model-free algorithms, and is one of the most widely used deep reinforcement learning algorithms for resource allocation problems.

Figure 3.3 shows the structure of a Deep Q Network. Similar to the structure of a Deep Neural Network (DNN), it consists of input, hidden, and output layers. The input layer takes state as input. The hidden layers perform nonlinear transformations

on the input data, enabling the network to capture relationships between states and actions. The output layer gives a vector of Q-values for each action, which is the expected cumulative rewards for taking different actions in this given state. It's worth notice that the Q-value of a state-action pair is an estimate of reward agent can expect to receive from that state onward, assuming it follows the optimal policy thereafter. The agent is then able to select the action with the highest Q-value to maximize its expected cumulative reward.

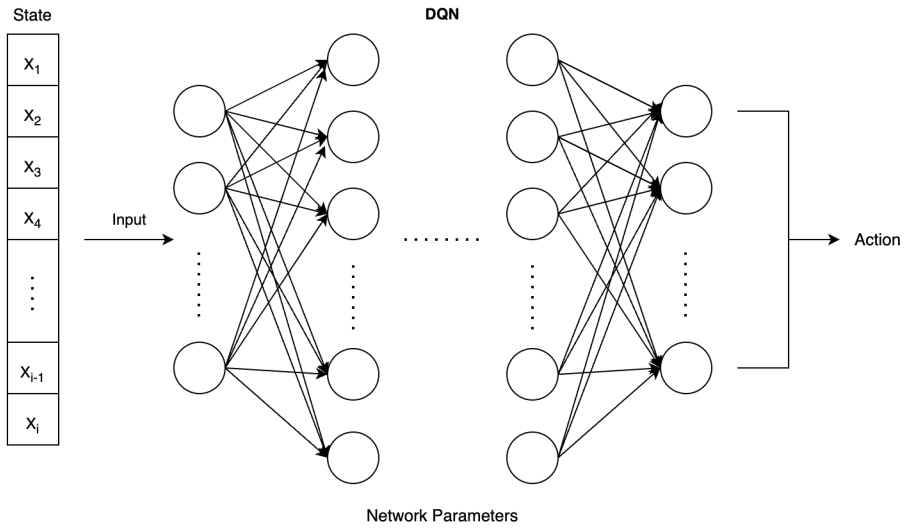

Figure 3.3: Deep Q Network

The architecture of DQN used in this thesis is described in table 3.1. We adopted a feedforward neural network with two fully connected layers as hidden layers. Both hidden layers have 120 neurons. As the input state is not structured data, nor is it grid-based, we used fully connected layer as input layer, so is the output layer. The input and output layers have 240 and $m * n$ neurons respectively, where m is number of power levels and n is number of resource blocks. The dimension of the output layer is the same as the range of actions the agent can choose from, and the DQN eventually outputs a Q-value for each possible action.

3.1.3 Double Q Learning

Double Q-learning (Double Q) adds a target Q network to DQN, it also falls in the category of value-based model-free algorithm. It mitigates the overestimation bias

Table 3.1: Description of DQN Architecture

Operation	Input Dimension	Output Dimension	Activation
Linear Layer	input_dim	240	ReLU
Linear Layer	240	120	ReLU
Linear Layer	120	120	ReLU
Linear Layer	120	$m * n$	ReLU

encountered in DQN, and provides a built-in mechanism to avoid sub-optimal decisions in resource allocation tasks. With a main network and a target network, actions are alternately selected based on the estimates from each function. Eventually this could lead to a more robust and reliable resource allocation strategy that outperform other algorithms.

The structures of the main Q-network and target Q-network in double Q are the same as described in table 3.1. A diagram illustration on the architecture of double Q learning is provided in figure 3.4.

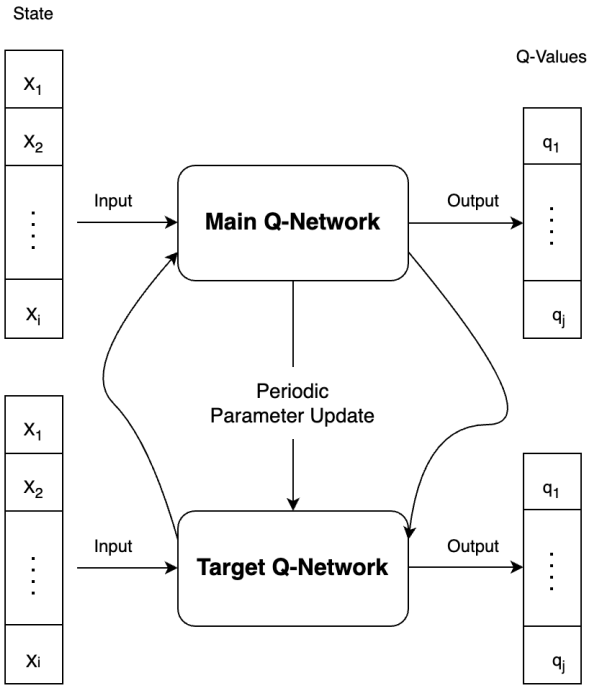

Figure 3.4: Double Q Network

3.2 System Model

In chapter four, we will compare the performance of three types of methods: Greedy approach with deep Q-learning, this is referred to as baseline performance; 4G setting with double Q-learning; 5G setting with double Q-learning with the proposed ICM-DQRA algorithm.

Algorithm 1 Baseline DQN Algorithm

```
Initialize the Q-network with random weights and biases
Initialize a memory buffer
for each epoch e do
    Initialize the environment with a random policy π
    Generate vehicles
    for each time step t do
        for each V2V link l do
            Get state s_t from the environment
            Take action a (selection of spectrum bands and transmission power) based
            on policy π
            Calculate reward r and move to next state s_{t+1}
            Append {s_t, a, r} to memory
        end for
        for each update step u do
            Sample a mini-batch from memory
            Update the policy π to maximize Q-value
        end for
        Update the weights and biases of the Q-network
    end for
end for
return Trained deep Q-network
```

In this section, we outline the workflow of these mentioned algorithms. Algorithm 1 outlines the baseline DQN algorithm.

In this baseline DQN algorithm, the agent chooses actions that maximizes Q-values without exploration. Additionally, in each update step, the policy π is updated to maximize Q-value, reinforcing the existing greedy behavior rather than exploring new actions. Hence the algorithm is considered greedy.

Additionally, in DQN, the agent uses a single Q-network to estimate the Q-values

for each action in a given state. Therefore, during training, the agent updates the Q-values through temporal difference learning where the Q-values are adjusted towards the target Q-values. Because the same Q-network is used for selecting actions and computing target Q-values, there is a risk that the Q-values may be overestimated.

Algorithm 2 Training Algorithm: Double Q-learning

```
Initialize a main and target Q-network Q and Q' with random weights and bias
Initialize a memory buffer
for each epoch e do
    Initialize the environment with random policy π
    Generate n vehicles
    for each time step t do
        for each V2V link l do
            Get state s_t from the environment
            Take action a based on policy π
            Calculate reward r and move to next state s_{t+1}
            Append {s_t, a, r} to memory
        end for
        for each update step u do
            Sample a mini-batch from memory
            Calculate target Q-values through the target network with equation (2.5)
            Update the main network using the sampled mini-batch and target Q-values with equation (2.6)
        end for
        update the weights and biases of the Q' with the weights and biases of Q periodically
    end for
end for
return Trained Q-network Q
```

To mitigate this, we apply double Q learning algorithm, outlined in algorithm 2, that applies a second Q-network, referred to as target Q-network in the system model.

3.2.1 Motivation for Curiosity

The main idea behind general reinforcement learning problems is to maximize reward associated with the environment. Such kind of reward is commonly known as extrinsic reward. Given the fact that such reward function is usually hard coded, it might suffer from the problem of not being scalable [20]. Namely, hard coded reward functions

are inherently inflexible. They may not capture the full complexity of the underlying problem or provide sufficient guidance for the agent to learn robust and adaptive behaviors. Consequently, relying solely on a hard-coded reward function can cause scalability issues by constraining the agent's ability to learn and generalize across diverse environments. The idea of intrinsic motivation is a solution among others, this gives rise to curiosity-driven learning.

The motivation for curiosity-driven learning is to build a reward function that is intrinsic to the agent itself. That is, to use a reward function generated by the agent itself. By doing this, the agent will become a self-learner. Such architecture assumes that if the agent can accurately predict the action that led from one state to another, it has implicitly learned a representation of the environment dynamics.

Another reason for adding the intrinsic reward to the agent is the exploration challenge that the agent faces. This challenge is cause by the state inputs to the double Q-learning algorithm.

To better visualise this problem, figure 3.5 shows a random state at time t, when a new task is generated:

```
tensor([ 0.7624, 0.6296, 0.6680, 0.6917, 0.8191, 0.5713, 0.7205,
      0.7438, 0.7279, 0.6736, 0.6660, 0.6411, 0.6516, 0.6966,
      0.5961, 0.8587, 0.6292, 0.7973, 0.6779, 0.7245, -1.0000,
      -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000,
      -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000,
      -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, 0.9258, 1.0163,
      0.9312, 1.0277, 0.9678, 0.9421, 0.8397, 0.9094, 0.8284,
      0.9131, 0.8807, 0.9206, 0.9840, 0.8012, 0.9646, 0.9778,
      0.9077, 0.8522, 1.1030, 0.9973, 1.0000, 0.0000, 0.0000,
      0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
      0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
      0.0000, 0.0000, 0.0000,1.000, 1.0000], device='cuda:0')
```

Figure 3.5: Sample State of a V2V Link at Time t

Specifically, this sate is made up of 6 parts:

1. Channel information from the V2V transmitter to BS (V2I link) at current time step t: $\mathbf{I}_{\mathrm{t}} = \{0.9258, 1.0163, 0.9312, 1.0277, 0.9678, 0.9421, 0.8397, 0.9094,$

0.8284, 0.9131, 0.8807, 0.9206, 0.9840, 0.8012, 0.9646, 0.9778, 0.9077, 0.8522, 1.1030, 0.9973}

2. V2V Communication Interference in previous time step $t-1$: $\mathbf{F}_{\text{t-1}}$ = {-1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1., -1.}

3. The instantaneous channel information of this V2V link at current time step t: $\mathbf{V}_{\text{t}}$ = {0.7624, 0.6296, 0.6680, 0.6917, 0.8191, 0.5713, 0.7205, 0.7438, 0.7279, 0.6736, 0.6660, 0.6411, 0.6516, 0.6966, 0.5961, 0.8587, 0.6292, 0.7973, 0.6779, 0.7245}

4. The selected sub-channels in previous time step $t-1$: $\mathbf{N}_{\text{t-1}}$ = {1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}.

5. Remaining time until exceed the time constraint: $T_t = 1$.

6. Remaining load that the vehicle still needs to transmit: $L_t = 1$.

Additionally, Figure 3.6 shows the next state of this V2V link at time $t+1$.

```
tensor([0.7624, 0.6296, 0.6680, 0.6917, 0.8191, 0.5713, 0.7205,
      0.7438, 0.7279, 0.6736, 0.6660, 0.6411, 0.6516, 0.6966,
      0.5961, 0.8587, 0.6292, 0.7973, 0.6779, 0.7245, -1.0000,
      -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000,
      -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000,
      -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, 0.9258, 1.0163,
      0.9312, 1.0277, 0.9678, 0.9421, 0.8397, 0.9094, 0.8284,
      0.9131, 0.8807, 0.9206, 0.9840, 0.8012, 0.9646, 0.9778,
      0.9077, 0.8522, 1.1030, 0.9973, 1.0000, 0.0000, 0.0000,
      0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
      0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
      0.0000, 0.0000, 0.0000, 0.9000, 0.6870], device='cuda:0')
```

Figure 3.6: Sample State of the Same V2V Link at Time $t+1$

We can see that the channel selection information of previous state is not changing, nor is the V2V communication interference. This is largely due to the nature of the problem we are solving: the environment is updated every 0.1 second, which is too short for any communication to finish, or for any vehicle information to change

significantly. Therefore, it is not common to see the state of a specific V2V link to change dramatically in any two consecutive time steps. As a matter of fact, we observed from a lager sample of states that the agent has the incentive to select same or identical channels for transmission, even though in this example, there are $N_{RB} = 20$ resource blocks available.

To make things worse, we also observed that the channel information of both the V2I and V2V links remain unchanged during the two time steps. The natural question we would ask is, should we model the agent's state this way, or in other words, is there a better way to model the state information, so that all fields of information are more relevant? The short answer is no. And the reason is that all of the existing fields, i.e., the V2V, V2I channel information, V2V communication interference, and the selected sub-channels are the environmental features that are relevant to the agent itself, and will have an impact on the agent's interactions with the environment.

This eventually gives rise to the exploration challenge to our agent. Namely the agent is less likely to explore alternative actions even if that might lead to a different, potentially better outcome. This exploration challenge limits the ability of the agent to find an optimal policy.

Most existing methods [5] [21] attempt to solve the problem from a data engineering point of view. However, in our example, we already observed that the state information is descriptive and relevant to the agent. This means that there isn't much to do about the state representation, and those methods are less likely to change the outcome for the better. Therefore, we propose to use the intrinsic curiosity module (ICM), which can be viewed as a built-on module to any given reinforcement learning algorithm, to address the problem.

We use double-Q learning algorithm to visualize the improvements that intrinsic rewards can bring.

3.2.2 The Intrinsic Curiosity Module (ICM)

Figure 3.7 shows the inner structure of an ICM agent.

The inputs to the ICM module are: agent's action at current time step t, agent's state at current time step t and next time step t+1. At current time step t, the agent would simulate the execution of the selected action in the environment, obtaining the

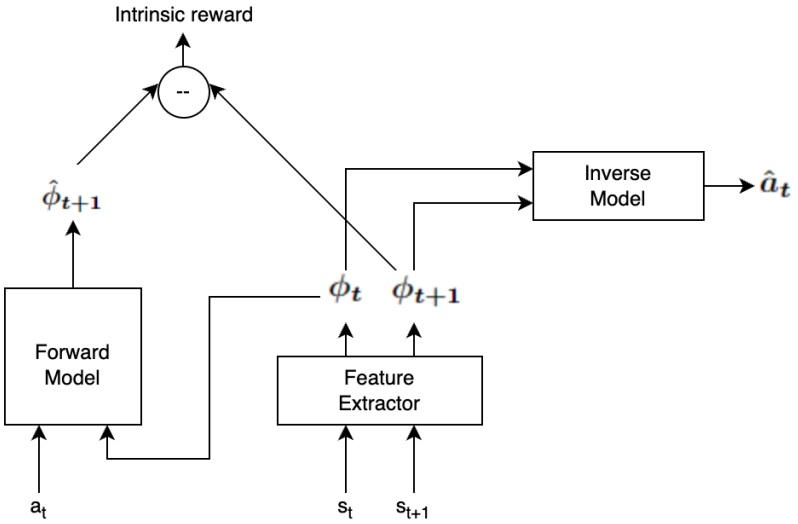

Figure 3.7: ICM Structure

actual next state.

Then, ICM encodes s_t and s_{t+1} into feature representations ϕ_t and ϕ_{t+1} via the feature extractor model. These feature representations are trained to predict agent's action at current time step $\hat{a}_t$ via the inverse dynamics model. The forward model of ICM takes agent's actual action at current time step a_t as input, along with the feature representation ϕ_t to predict the feature representation $\hat{\phi}_{t+1}$ of s_{t+1}, the state representation at next time step.

The prediction error of the feature representations is used as the curiosity based intrinsic reward. Namely, the intrinsic reward r' is calculated as:

$$r' = \eta * 0.5 * (\hat{\phi}_{t+1} - \phi_{t+1})^2, \tag{3.1}$$

where η is a scaling factor. Generally, a higher value of η encourages the agent to prioritize exploration, as more emphasize is given to intrinsic reward. During simulation, we set $\eta = 2$ after parameter tuning.

The training procedure of the ICM agent is outlined in algorithm 3.

Figure 3.8 shows how the ICM agent can fit into the double Q-learning algorithm [18]: at time step t, the agent interacts with the environment by taking action a_t based on the policy π, and transition to state s_{t+1}. The policy π is trained to maximize the sum of extrinsic reward r_t and the intrinsic reward r'_t generated by the ICM agent. Now that the agent is at time step $t+1$, it will repeat the same process at

Algorithm 3 Training the ICM Agent with Mini-Batch

for each epoch e **do**
  Shuffle the memory
  for each mini-batch m from memory **do**
    for each entry i in the mini-batch **do**
      Extract feature representations ϕ_t^i and ϕ_{t+1}^i for states s_t^i and s_{t+1}^i
      Predict the action $\hat{a}_t^i$ from feature representations through the inverse model
      Extract the predicted feature representation of next state $\hat{\phi}_{t+1}^i$
    end for
    Calculate the MSE loss between a_t^i and $\hat{a}_t^i$
    Update network parameters to minimize the error
  end for
end for
return Trained ICM agent

it did in time step t. This iterative process will move on with the training algorithm, providing intrinsic reward to the double Q-learning algorithm at each time step.

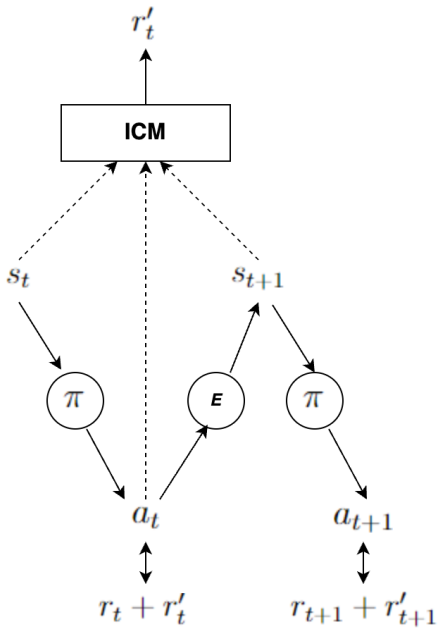

Figure 3.8: ICM Workflow

3.2.3 Proposed ICM-DQRA Method

The ICM-DQRA algorithm we proposed for this thesis is an ICM-based learning approach. The ICM agent is composed of three parts: the forward model, the feature extractor, and the inverse model. Algorithm 4 outlines the training algorithm for ICM-DQRA.

Algorithm 4 Training Algorithm: ICM-DQRA

```
Initialize main and target Q-networks Q and Q' with random weights and biases
Initialize a memory buffer
for each epoch e do
    Initialize the environment with random policy π
    Generate n vehicles
    for each time step t do
        for each V2V link l do
            Get state s_t from the environment
            Take action a based on policy π
            Calculate reward r and move to next state s_{t+1}
            Load ICM agent and calculate intrinsic reward r'
            Update reward r = r + r'
            Append { s_t, a, r, s_{t+1}} to memory
        end for
        for each update step u do
            Sample a mini-batch from memory
            Calculate target Q-values through the target network with equation (2.5)
            Update the main network using the sampled mini-batch and target Q-
            values with equation (2.6)
        end for
        update the weights and biases of the Q' with the weights and biases of Q
        periodically
    end for
end for
return Trained Q-network Q
```

Though the information of next state is not required for the double Q-learning algorithm, it needs to be included in the replay memory to facilitate the training and execution of the ICM agent.

The nature of the ICM agent is a built-on module to any given existing reinforcement learning algorithm. Therefore once trained, it can be adapted to the changing

environment and the only thing we need to change is the input dimension, which should always be equal to the dimension of the state space.

Finally, the testing algorithm that generates simulation results is provided in algorithm 5.

Algorithm 5 Testing Algorithm

```
Load the Q-network model
for each epoch e do
    Generate vehicles
    for each time step t do
        Create an action buffer A
        for each V2V link l do
            Get state s_t from the environment
            Choose action a_t with maximum Q-value with s_t as input from the Q-network model
            Append a_t to A
        end for
        Interact with the environment based on A
    end for
end for
Calculate the sum rate of V2I links
Calculate the probability of satisfied V2V links
Calculate the average transmit power of V2V links
return Simulation results
```

With these, we will be able to deploy the ICM-DQRA algorithm into participating vehicles. Those vehicles are typically equipped with onboard sensors, processors, and communication modules, and are capable of making decisions as well as interacting with the environment in real-time. The vehicle that acts as the sender of a packet will create the V2V transmission link and execute the algorithm, make the decision with respect to which power level to choose for transmission. This distributed approach can help reduce latency, enhance privacy, and increase scalability when compared to centralized approaches where the algorithms are implemented in base stations instead. However, the bases stations will serve the purpose of policy distribution and updates. This allows for coordinated learning across the vehicles and ensures that all vehicles have access to the latest policies.

Before proceeding with our evaluations, we need to have a well-defined ICM agent.

After conducting various tests with 40 vehicles in the simulation area, we present the final structure of the ICM agent from table 3.2 to table 3.4. Two types of deep learning models: RNN and MLP are adopted.

The feature extractor is a RNN-based model. It takes states as input, therefore the size of input is the same as the length of the state vector. We used four recurrent layers with each layer processing input data sequentially. The value of the parameter *hidden_size* is the same as the size of the feature representation vector ϕ in figure 3.7.

Table 3.2: Structure of RNN Based Feature Extractor

Operation	**Input Dimension**	**Output Dimension**	**Activation**
Recurrent Layer	input_dim	hidden_size	ReLU
Recurrent Layer	hidden_size	hidden_size	ReLU
Recurrent Layer	hidden_size	hidden_size	ReLU
Recurrent Layer	hidden_size	hidden_size	ReLU

Within each recurrent layer, the activation function employed is Rectified Linear Unit (ReLU), defined as follows:

$$ReLU(x) = \max(0, x). \tag{3.2}$$

The motivation for choosing ReLU activation function is to introduce non-linearity to the network, allowing it to effectively learn feature representations from the input states.

The inverse model is MLP based, designed to predict actions based on feature representations. The first linear layer takes as input a concatenation of two feature representations, each of size *hidden_size*, therefore the input size is $hidden\_size * 2$. This layer maps the concatenated input to a new representation with a dimension of *hidden_size*. Subsequently, another linear layer processes this intermediate representation. Finally, the output layer produces an action prediction, with the output dimension equal to the number of actions. In this case, this value equals to 1 because the action is one numerical number indicating the agent's choice of spectrum and power level.

We only apply a Sigmoid activation function to the output layer that maps any

Table 3.3: Structure of MLP Based Inverse Model

Operation	Input Dimension	Output Dimension	Activation
Linear Layer	hidden_size * 2	hidden_size	None
Linear Layer	hidden_size	hidden_size	None
Linear Layer	hidden_size	n_actions	Sigmoid

number to a value between 0 and 1:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3.3}$$

The Sigmoid activation function will effectively normalize the agent's action, and this will make sure that in case the available resource blocks or power levels to the agent changes, the agent's action space will change, but the values of a and $\hat{a}$ will still be in the same range from 0 to 1.

The MLP-based forward model aims to predict future representation for next state with current state representation and action. It consists of three linear layers. The inputs to the first layer are action a_t and state representation ϕ_t, therefore the input size is $hidden\_size + 1$. The output dimension of the first layer, input and output dimension of the second layer, and the input dimension of the third layer are all set to $hidden\_size/2$. The task of this model is to predict feature representation $\hat{\phi}_{t+1}$, therefore the output dimension of the third layer is the same as $|\phi_{t+1}|$, which is $hidden\_size$. We only apply ReLU activation function to the last layer.

Table 3.4: Structure of MLP Based Forward Model

Operation	Input Dimension	Output Dimension	Activation
Linear Layer	hidden_size + 1	hidden_size/2	None
Linear Layer	hidden_size/2	hidden_size/2	None
Linear Layer	hidden_size/2	hidden_size	ReLU

3.3 Problem Formulation

In the V2V communication scenario, an agent is essentially a V2V link. Even though there are multiple agents (multiple V2V links) in a communication network, they operate independently. Therefore the whole system is considered to be a single-agent setting from the perspective of reinforcement learning. In each iteration, the agent

selects an action that maximizes the Q-value calculated from the target Q-network. The equations we use to update the Q-values are the same as described in equations (2.6) and (2.7).

The relationship between Q-values and reward is that Q-values represent the expected cumulative reward for taking some action a in state s, whereas the reward itself, obtained from the reward function, provides the immediate feedback to the agent for taking some action a in state s. They jointly guide the agent's decision-making process and finding the optimal policy $\pi*$.

In this section, we formally define the resource allocation problem as MDP.

3.3.1 Action

The action space (A) contains a collection of all possible actions that the agent can take. An action (a) for an agent is a selection of transmission power and spectrum resource with $a \in A$. Let m be the number of power levels and n be the number of resource blocks. In our case the resource block is considered as a portion of the spectrum allocated for data transmission during a specific time interval. Consequently, we use an integer within the range 0 to mn to represent the action a taken at any time t:

$$\{a_t \mid 0 \leq a_t < mn, a_t \in \mathbb{N}\}. \tag{3.4}$$

Specifically, if we have 20 resource blocks and three power levels for transmission, we would use an integer between 0 and 59, inclusive to represent any possible action that the agent can take.

3.3.2 State

The state space (S) provides relevant information needed for the agent's decision-making. We represent the state of an agent in six parts:

1. $\mathbf{I_t}$: Channel information from the vehicle to BS (V2I link) at current time step t. Specifically, for a given V2I channel, its channel information is modeled from the pass loss and shadowing:

$$\mathbf{I_t} = P_t + S_t, \tag{3.5}$$

where I_t represents V2I channel information, P_t denotes path loss, which measures signal attenuation that captures the difference between transmitted and received power levels.

Previous research [14] indicates that path loss can be effectively modeled using a logarithmic function of distance between the transmitter and receiver:

$$P_t = A + 10 \cdot n \cdot \log_{10}\left(\frac{\sqrt{d_1^2 + d_2^2 + (h_{\text{bs}} - h_{\text{ms}})^2}}{1000}\right), \tag{3.6}$$

where d_1 and d_2 represents the difference between the x- and y-coordinate of the vehicle to the base station (BS), respectively, h_{bs} and h_{ms} are the heights of the base station and mobile station (the vehicle in this case), respectively. A and n represent intercept and path loss exponent, respectively.

Shadowing, on the other hand, is usually caused by any obstacles between the transmitter and receiver, and previous research [14] has shown that the shadowing effect can be effectively modeled with an exponential decay function:

$$S_t = \exp\left(-\frac{d_{ij}}{d_0}\right) \cdot S_{t-1} + \sqrt{1 - \exp\left(-2 \cdot \frac{d_{ij}}{d_0}\right)} \cdot N_0. \tag{3.7}$$

Here d_{ij} represents the Euclidean distance between the receiver and the transmitter, and d_0 is the decorrelation distance represents a threshold over which the V2V channels become decorrelated and independent from each other.

Apart from that, N_0 represents a random sample from a normal distribution:

$$N_0 \sim \mathcal{N}(0, \delta^2), \tag{3.8}$$

where δ^2 is the standard deviation of shadow fading.

2. $\mathbf{F}_{\text{t-1}}$: V2V communication interference in previous time step $t-1$. The formula used to calculate the overall interference for all V2V links sharing the same resource block is:

$$\mathbf{F}_{\text{t-1}} = 10^{\frac{P_{\text{V}} - P_{\text{t-1}} + G}{10}} + 10^{\frac{P_N}{10}}, \tag{3.9}$$

where P_V is the transmit power level in dBm, P_{t-1} denoted the pass loss between the transmitter and the receiver, measure in decibels. P_N is the noise power,

and is converted to its linear scale in milliwatts, G is the combined antenna gain and receiver noise figure that captures the overall performance in terms of signal reception and noise handling, calculated through:

$$G = 2 \times V_G - V_N, \tag{3.10}$$

where V_G is the vehicle antenna gain, V_N is the vehicle noise figure.

The obtained interference value is effectively scaled within the range of -1 and 1, and the goal of the proposed resource allocation method is to minimize the interference while satisfying the latency constraints.

3. $\mathbf{V}_{\text{t}}$: The instantaneous channel information of this V2V link at current time step t. Similar to the V2I channel, the channel information of a given V2V channel is also modeled using log-distance model as follows:

$$V_t = P_t + S_t, \tag{3.11}$$

where V_t is V2V channel information, P_t denotes path loss, S_t represents shadowing.

The pass loss and shadowing are calculated through equations (3.12) and (3.13), respectively:

$$P_t = L_0 + 10 \cdot n \cdot \log_{10}\left(\frac{d_{ij}}{d_0}\right), \tag{3.12}$$

$$S_t = \exp\left(-\frac{d_{ij}}{d_0}\right) \cdot S_{t-1} + \sqrt{1 - \exp\left(-\frac{d_{ij}}{d_0}\right)^2} \cdot N_0, \tag{3.13}$$

where L_0 is the path loss at the reference distance $d = d_0$.

4. $\mathbf{N}_{\text{t-1}}$: The selected sub-channels in previous time step $t-1$. For any value $\rho_k \in \mathbf{N}_{\text{t-1}}$, if $\rho_k[m] = 1$, this is an indicator that means the kth channel is selected to carry out the communication and $\rho_k[m] = 0$ means otherwise.

5. T_t: Remaining time until exceed the time constraint. In the simulation, we aim to deal with time-sensitive tasks such as navigation services, emergency information sharing, etc. The time constraints (denoted by T_0, and represents the total time allocated for a task to finish) for every task is the same during the

simulation. In the simulation, we tested each task with their individual time constraint T ranging from 100 milliseconds (ms) to 1 second (1000ms), with an interval of 100ms. Additionally, the entire traffic network is updated every 100ms, this includes vehicle information, task load, remaining time, etc.

6. L_t: Remaining load that the vehicle still needs to transmit. This value is also used to sort the tasks so that we are able to prioritize tasks with higher load. Initially, all newly created tasks have a remaining workload value 1 (100%), this value would gradually decrease for the same task until fulfilled or eventually exceed the time constraint, resulting in a failure. During each network update, this value is updated to be the proportion of bits still need to be transmitted using the equation below:

$$L_t = \frac{T_i}{T_0}, \tag{3.14}$$

where T_i is the remaining time needed to complete the transmission.

The final state $\mathbf{S}_\mathrm{t}$ of a given V2V link is thereafter the concatenation of the above six components:

$$\mathbf{S}_\mathrm{t} = \{\mathbf{I}_\mathrm{t}, \mathbf{F}_\mathrm{t\text{-}1}, \mathbf{V}_\mathrm{t}, \mathbf{N}_\mathrm{t\text{-}1}, T_t, L_t\}. \tag{3.15}$$

The state transition from s_t to s_{t+1} with reward r_t after taking action a_t is governed by the conditional transitional probability:

$$p(s_{t+1}, r_t \mid s_t, a_t) \tag{3.16}$$

that encapsulates the stochastic nature of the environment, which refers to the uncertainties in the outcomes of state transitions and rewards within the environment. The agent, on the other hand, has no prior knowledge of the transition probabilities and must learn from its interactions with the environment.

3.3.3 Reward

The reward function (R) calculates the immediate benefit or cost while taking action a in state s. The reward function in this thesis is jointly determined by the capacities of the V2V and V2I links, as well as the time spent for transmission (latency constraints). Equation (3.17) gives the mathematical definition of the reward function:

$$r_t = \lambda_1 \sum_i C^c[i] + \lambda_2 \sum_j C^v[j] - \lambda_3 (T_0 - T_t). \tag{3.17}$$

In this equation, $\sum_i C^c[i]$ represent the sum capacity of V2I links where $C^c[i]$ represent the capacity of the ith V2I link. And $\sum_j C^v[j]$ denote the sum capacity of V2V users with $C^v[j]$ representing the capacity of jth V2V user. The term $T_0 - T_t$, which is the time constraint minus remaining time until exceed the constraint, represents the time spent for transmission; λ_1, λ_2, and λ_3 represent the weights of the three components in the reward function with $\lambda_1 + \lambda_2 = 1$, and λ_3 to be a value close to λ_1. After conducting some sets of testings to the parameters, we set $\lambda_1 = 0.1$ for simulation, so that the reward function is mostly dominated by the capacity of V2V links, while at the same time take into account of the other two factors.

To calculate the capacity of cellular users, we use the following equation:

$$C^c[i] = W \cdot \log(1 + \gamma^c[i]), \tag{3.18}$$

where W is the bandwidth, and $\gamma^c[i]$ is the signal-to-noise ratio (SNR) of the ith V2I link. We assume that the communication channels are orthogonal to each other, which implies that the signals transmitted by different vehicles do not interfere with each other significantly. Specifically, $\gamma^c[i]$ is calculated through the equation below:

$$\gamma^c[i] = \frac{P_i^c h_i}{\sigma^2}, \tag{3.19}$$

where P_i^c represents the transmission power of the ith V2I link, h_i is the antenna gain of for this link, σ^2 is the noise power, P_j^v represents the transmission power of the jth V2V link, h_j is the antenna gain of the this link.

To calculate the capacity of V2V users, we use the following equation:

$$C^v[j] = W \cdot \log(1 + \gamma^v[j]), \tag{3.20}$$

where $\gamma^v[j]$ is the SNR of the jth vehicle, and is calculated through the equation below:

$$\gamma^v[j] = \frac{P_j^v h_j}{\sigma^2}, \tag{3.21}$$

where P_j^v represents the transmission power of the jth V2V link, g_j is the antenna gain of this link.

The objective of the reinforcement learning algorithm is to maximize the discounted cumulative reward. Specific values of transmit power levels, noise powers, noise figures, and antenna gains used for simulation are given in the parameter table 4.1 in chapter 4.

Chapter 4

Performance Analysis

In this chapter, we first introduce the simulation environment we used, which is an urban simulation area defined by the 3rd Generation Partnership Project (3GPP). Additional information related to vehicle generation, the agent, and training of the ICM agent are provided. Before showing our results, we depict the experiment setup and the evaluation metric used to evaluate our results. Finally, we present our testing results for both non-ICM and ICM-Based schemes with a thorough discussion.

4.1 Simulation Environment

The simulation environment is an urban case defined following the 3GPP TR36.885 V2.0.0 standard [1]. Figure 4.1 provides an illustration of the experiment scenario for this thesis.

The height and width of the simulation area are 1299 meters and 750 meters respectively. The original point is located at the button-left corner with coordinate (0,0) and the base station is placed at the center of the simulation area, whose coordinate is (649.5, 375), and the transmission power is 36dBm [6]. Additionally, each street is made up of four lanes (two lanes in each direction), and the width of each lane is 3.5 meters. Both sparse and dense traffic scenarios are taken into consideration during simulation.

Table 4.1 provides the parameters used in simulation.

In the simulation environment, vehicles are added randomly to emulate realistic vehicular movement patterns. Specifically, three parameters are given while adding a vehicle: starting position, direction, and speed. The starting positions for each vehicle along predefined lanes are randomly chosen, the direction of a vehicle includes down, up, left, and right, this direction represents a vehicle's initial movement trajectory. Then, each vehicle is assigned a random velocity in the range 10 to 15 with equal probability to reflect varying speeds among vehicles. The number of vehicles

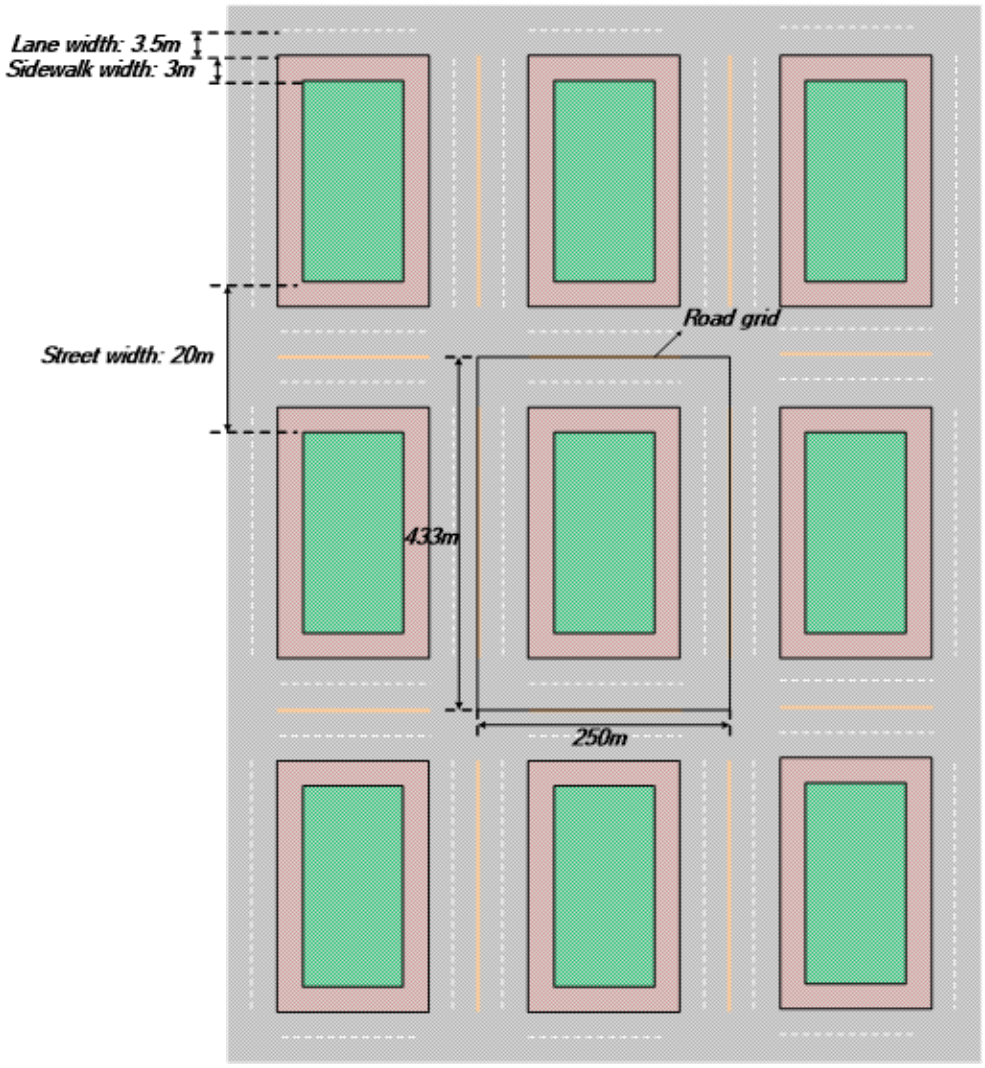

Figure 4.1: Experiment Scenario

Table 4.1: Parameter Table

Parameter	**Value**
Resource Blocks	20
Intercept	128.1
Pass Loss Exponent	3.76
Standard Deviation of Shadow Fading δ^2	8 dB
Decorrelation Distance d_0	50 m
Vehicle Transmit Power Levels	[5, 10, 23] dBm
Base Station Power Level	36 dBm
Noise Power	-114 dBm
Carrier Frequency	2 GHz
Bandwidth	4 GHz
Height of Vehicles	1.5 m
Vehicle Antenna Gain	6 dBi
Vehicle Noise Figure	9 dB
Vehicle Speed	10 - 15 m/s
Height of Base Stations	25 m
Base Station Antenna Gain	10 dBi
Base Station Noise Figure	4 dB

to be added into the environment is a parameter specified before the start of each simulation.

To better reflect real-world settings, shadowing effects are simulated to model wireless communication scenarios. Specifically, Gaussian noise with specific standard deviations is applied to represent shadowing effects for both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication links. This ensures the creation of a more realistic environment that is suitable for evaluating the performance of vehicular communication systems. Moreover, we apply dynamic resource allocation techniques where resource blocks are allocated based on demand and network conditions. In this case, V2V and V2I links can temporarily share resource blocks and we employ interference management techniques such as power control to mitigate interference and ensure reliable communication.

4.2 Training the Intrinsic Curiosity Module

Following algorithm 3, we perform training on our ICM agent. Figure 4.2 shows the Mean Squared Error (MSE) between the predicted action $\hat{a}_t$ and actual action a_t.

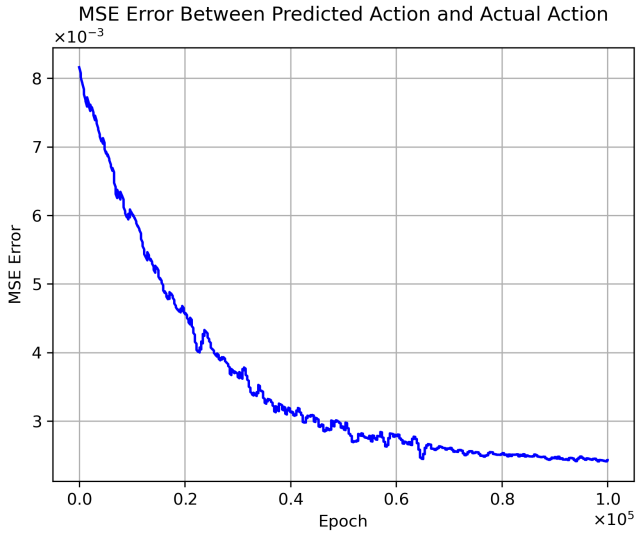

Figure 4.2: MSE Error Between Predicted and Actual Action

It is observed from the diagram that the MSE error between a_t and $\hat{a}_t$ has a decreasing trend from around 8.0×10^{-3} to less than 2.5×10^{-3}. The MSE error becomes stable after roughly 70,000 epochs, and remains stable thereafter. Therefore we conclude from the diagram that the ICM agent is able to learn a meaningful feature representation for input states. The trained ICM agent is then built into double Q-learning to provide an intrinsic reward to the agent.

The general idea behind the ICM agent is that, as long as the output of the inverse model, that is, predicted action at time step t, is "similar" to the actual action at time step t, given as the input to the forward model, we know that the ICM agent learns a meaningful feature representation of the input states. We measure this "similarity" using the MSE error between $\hat{a}_t$ and a_t. Once we have finished training this ICM agent, it can be viewed as a built-on module to any given reinforcement learning algorithm, whether it is double Q-learning, or deep Q-learning, etc. We only need to set a flag for ICM and load it to calculate the intrinsic reward for us. Another advantage of this approach is that during simulation, we verified that this agent does not need to be re-trained if we were to change certain parameters, such as number of vehicles during simulation. Though it might take a relatively long time to train the ICM agent, this advantage would save time for us in future simulations.

4.3 Experimental Setup and Evaluation Criteria

We use PyTorch, an open source machine learning framework to build the proposed architectures. In the simulation, both sparse and dense traffic scenarios in urban areas are considered. Specifically, we examined the simulation area with 20, 40, 60, 80, and 100 vehicles. Unless otherwise specified, all other parameters in Table 4.1 remain the same across all simulations.

We employed the following metrics to evaluate the performance of our proposed ICM-based resource allocation scheme:

1. Probability of Satisfaction: This metric measures the probability that V2V links in the simulation area satisfy a predefined latency constraint. A higher probability of satisfaction indicates a greater proportion of V2V links successfully meeting the latency requirements, as well as the effectiveness of the proposed resource allocation scheme.

2. V2I Sum Rate: This metric evaluates the aggregate data rate achieved by V2I links in the simulation area. It represents the total throughput or capacity of V2I communication within the network. Though a higher V2I sum rate indicates greater data transmission capacity, it is usually difficult to achieve because the interference between V2V links and V2I links will grow as more vehicles are added during simulation.

3. Power Level Selection: This metric assesses the distribution of power levels selected by vehicles for communication, which is the choices made by vehicles with respect to which transmission power levels to use for communication. This selection of power level will have a direct impact on interference and energy consumption.

These metrics collectively provide insights into the performance and effectiveness of the proposed resource allocation scheme. They also provide insights into the proposed scheme's ability to meet communication requirements, utilizing various resources, and enhancing overall network performance for the vehicular communication system.

4.4 Experimental Results and Discussions

The proposed ICM-DQRA resource allocation scheme is compared with the following resource allocation schemes:

1. Greedy (baseline) solution: described in algorithm 1, this method uses DQN for resource allocation. It is considered greedy due to the update rules to the policy π. The policy is updated to maximize Q-value, reinforcing the greedy behavior rather than exploring new actions.

2. Double Q solution in 4G setting: Given the fact that a significant number of mobile devices still use LTE technology, we perform simulation on 4G setting. Table 4.2 illustrates the updated parameters used in 4G simulation compared with Table 4.1.

Table 4.2: Updated Parameters in 4G Setting

Parameter	**Value**
Carrier Frequency	850 MHz
Bandwidth	20 MHz
Base Station Antenna Gain	5 dBi
Base Station Noise Figure	2.5 dB

3. Double Q solution in 5G setting: This method is described in algorithm 2, same as the method used in 4G setting. The improvements that an intrinsic reward brings is visualized by its simulation results and the simulation results for ICM-DQRA.

4.4.1 Probability of Satisfaction

Figure 4.3 shows the probability of V2V link failure rate versus the number of vehicles. A failed transmission indicates that the V2V link is not able to satisfy the pre-defined latency constraint.

In general, the failure rate shows an increasing trend. The reason is with the increase of vehicles, the number of V2V links in the communication system also increases. As a result, the communication will become more competitive, making it harder to satisfy the latency constraints for all vehicles.

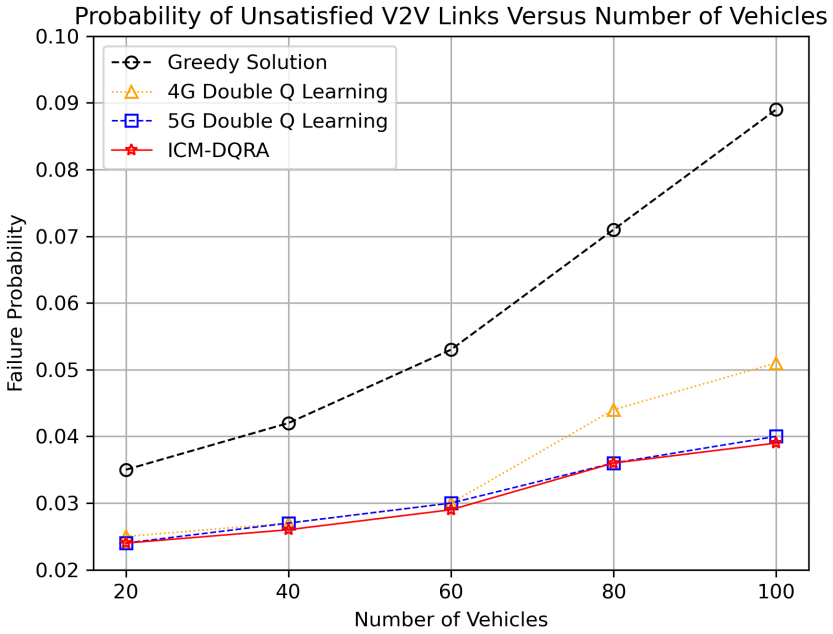

Figure 4.3: V2V Link Failure Rate Versus Number of Vehicles

An interesting observation is that the failure rate is identical between 4G and 5G settings when there are fewer vehicles, but the rate increases significantly when the traffic becomes dense. This is mainly caused by the nature of 4G versus 5G, and the different sets of parameters used during simulation. The lower carrier frequency and narrower bandwidth in the 4G setting result in an increase in interference compared to the 5G setting. Higher interference levels eventually degrade signal quality and cause likelihood of communication failures to increase, particularly in environments with a high density of vehicles.

Another observation is that the failure rate of double Q solution, whether in 4G or 5G, is identical to the failure rate of ICM-DQRA solution. But the failure rate of the baseline performance is significantly higher. This is mainly due to the advantage of double Q-learning.

In the reward function described in equation (3.17), the weight of the penalty term $\lambda_3(T_0 - T_t)$ is small so that more emphasis lies on the capacity of V2V links. The choice of weight parameters λ worked well in the reward function during simulation, but for DQN algorithm, it more or less facilitate the greedy choice of the agent due to the relatively small weight on the penalty term. Additionally, instead of greedily

choosing higher power levels for transmission and allocating more resource for all V2V links possible, double Q learning is able to dynamically adjust the power and spectrum for transmission so that the V2V links that are more likely to violate the latency constraints are allocated to more resources. This effectively separate the V2V links into different groups with different priority levels, and only those with higher priority are allocated to more resources.

We can also observe from the diagram that the failure rate of proposed ICM-DQRA algorithm is identical to the double Q-learning method. This indicates that with intrinsic reward, though the agent explored a larger action space and took action it wouldn't have taken in double Q-learning, it did not take actions randomly. Rather, the agent selects different power levels for transmission while satisfying the latency constraint at the same time.

4.4.2 V2I Sum Rate

V2I sum rate, also known as V2I capacity, is measured in megabits per second. It quantifies data transmitted from vehicles to infrastructure over a given period of time and reflects the combined capacity for data transmission from vehicles to nearby infrastructures, which in our case is a base station. Figure 4.4 shows the summation of V2I rate versus the number of vehicles.

From the figure, we can observe that with an increasing number of vehicles, the V2I capacity shows a decreasing trend, regardless of which algorithm we use, or which simulation setting it is.

This phenomenon is caused by several reasons. The main reason is because the number of V2V links grow when the number of vehicles increase, this causes the interference for V2I links to increase, thereby the V2I capacity drops. Collisions can also occur when multiple vehicles attempt to transmit data simultaneously, leading to packet loss, re-transmissions, etc. However, the proposed ICM-DQRA method still outperforms traditional double Q learning method, indicating that with intrinsic reward, it better mitigates the interference between V2V and V2I links.

Additionally, more vehicles indicate more data is transmitted simultaneously, leading to the communication channels being congested. This can not only lead to increased interference but also reduced signal quality, thereby impact the achievable

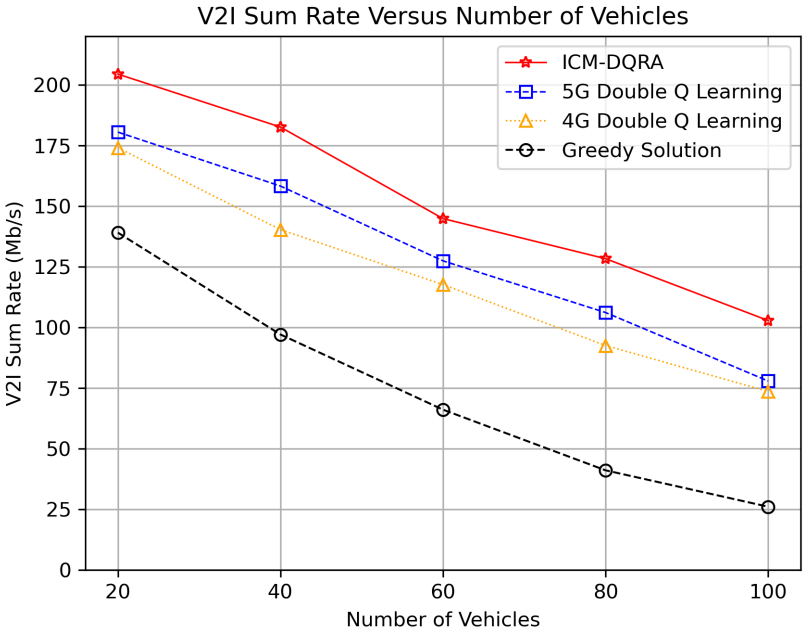

Figure 4.4: V2I Sum Rate Versus Number of Vehicles

data rates. Apart from congestion, more "competition" for accessing resources also occurs in dense traffic scenarios. The available bandwidth for V2I communication is finite and shared among vehicles, therefore vehicles will compete frequency bands and time slots provided by the base station, leading to increased contention and reduced throughput. Different resource allocation algorithms will have an impact on this criteria, and the ICM-DQRA algorithm mitigates this situation well.

One interesting observation is that the V2I sum rate in 5G setting only outperforms 4G setting by a little bit, the numerical values do not differ by a lot when the traffic becomes dense. This is in contrast with the failure rate in figure 4.3.

To conclude, the reverse relationship between V2I sum rate and the number of vehicles highlights the challenges associated with managing communication for dense traffic as well as the importance of efficient resource allocation algorithms for vehicular networks.

4.4.3 Power Level Selection

To mitigate interference and maintain communication reliability, vehicles may need to increase their transmit power levels. However, higher transmit powers indicate

higher energy consumption. And one bottleneck for 5G cellular technology is high energy consumption. Therefore it is crucial to find a balance between them.

In this sub-section, we study the agent's power selection behaviours at different time of transmission, and in different traffic densities, under 5G setting. The power selection results apply only to V2V links as the power level the base station operates is fixed.

In order to better visualize the effect of the ICM agent, we show two sets of results: one for double Q learning with out intrinsic reward, and another for the proposed ICM-DQRA method, which introduces intrinsic reward. Figure 4.5 shows the power selection behaviour when there are 20 vehicles in the simulation area. The diagram on the left is the power selection result for Double Q learning, and the diagram on the right is the power selection result for the proposed ICM-DQRA method.

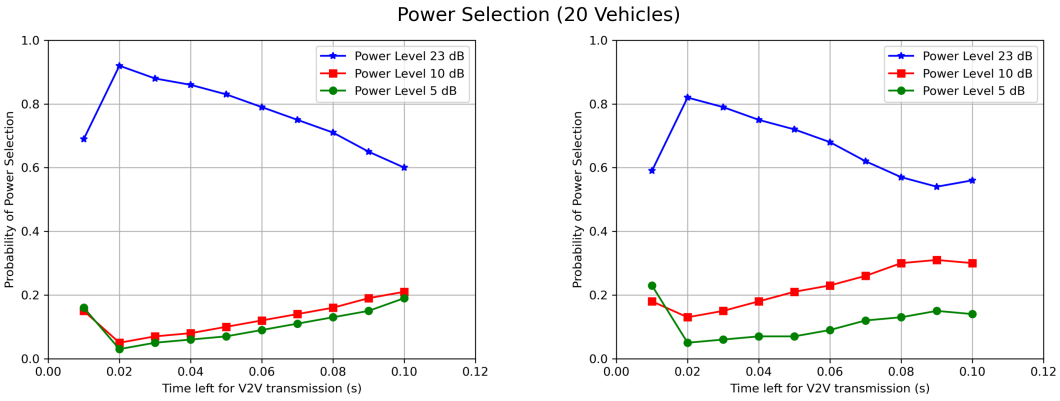

Figure 4.5: Power Selection without (Left) and with (Right) ICM (20 Vehicles)

We can observe from both diagrams that in general, the probability for an agent to choose high power level for transmission is low when there are sufficient time for transmission. As remaining time decreases, the agents have a tendency to choose higher power levels for transmission to ensure that the latency constraint is met.

The only exception to this pattern is when there is only 10ms left. At this point, the probability of the agent selecting the maximum power level dropped significantly because through training, the agent learned that either a task can be fulfilled with

a lower power level because it's been some time since the task is generated, or the agent realized that even if it occupies the maximum power level, the task cannot be fulfilled anyhow. Therefore through interactions with the environment, it learned to choose lower power levels to reduce interference between V2V and V2I links, this can also increase the reward the agent receives.

This power selection decision indicates that the double Q learning algorithm and the ICM-DQRA approach is able to capture the implicit relationship between the agent's state and the reward function.

When we compare both results, we notice that in ICM-DQRA method, the agent has a relatively low probability to choose the maximum power level at all times. Additionally, we notice a higher probability for the agent to choose the medium power level for transmission. This give a less aggressive resource allocation result, and is achieved by the intrinsic reward, which is absent in double Q learning.

As a next step, we add more vehicles to the simulation area, and study the performance of both algorithms under medium and dense traffic scenarios. Figures 4.6, 4.7, and 4.8 show the power selection result when there are 40, 60, and 80 vehicles in the simulation area respectively.

We can observe from these three sets of diagrams that as more vehicles enter the simulation area, the resource allocation task becomes more competitive. Both algorithms show an increasing trend with respect to the probability for the agent to choose the highest power level.

For double Q learning, its probability to choose the highest power level is already high, therefore apart from some minor difference, we do not see any significant change in the agents' behaviour when there are more vehicles.

Whereas for ICM-DQRA method, the probability for the agent to choose the highest power level also increases when more vehicles exist in the simulation area. But it still outperforms double Q learning with respect to energy consumption.

We also observed that in ICM-DQRA, the probability for the agent to choose the lowest power level increases as well, and becomes closer to the probability for the agent to choose medium power level. However in double Q learning, their probabilities are always identical. This shows that with intrinsic reward, the algorithm can effectively distinguish the needs of agents and assign the most appropriate power level for their

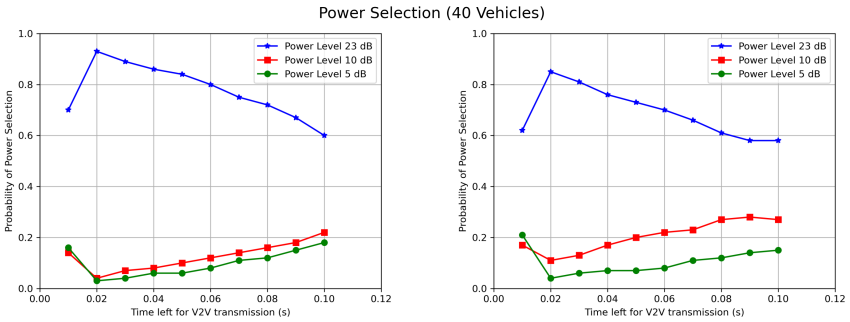

Figure 4.6: Power Selection without (Left) and with (Right) ICM (40 Vehicles)

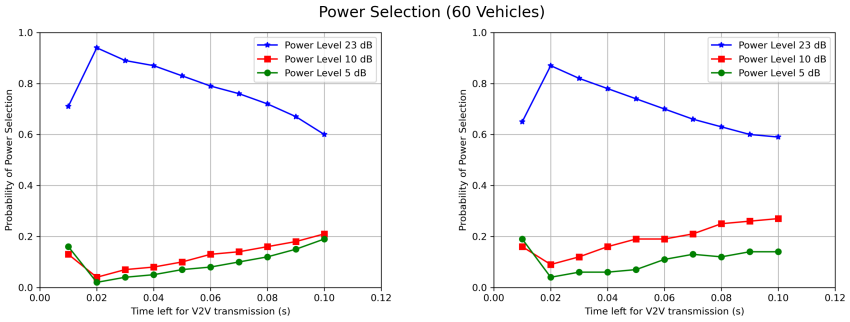

Figure 4.7: Power Selection without (Left) and with (Right) ICM (60 Vehicles)

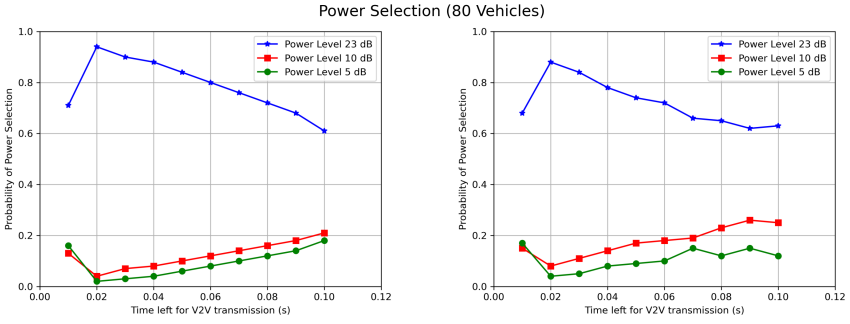

Figure 4.8: Power Selection without (Left) and with (Right) ICM (80 Vehicles)

transmission, eventually utilizing limited network resources.

Figure 4.9 shows the power selection results in dense traffic scenario where 100 vehicles enter the simulation area.

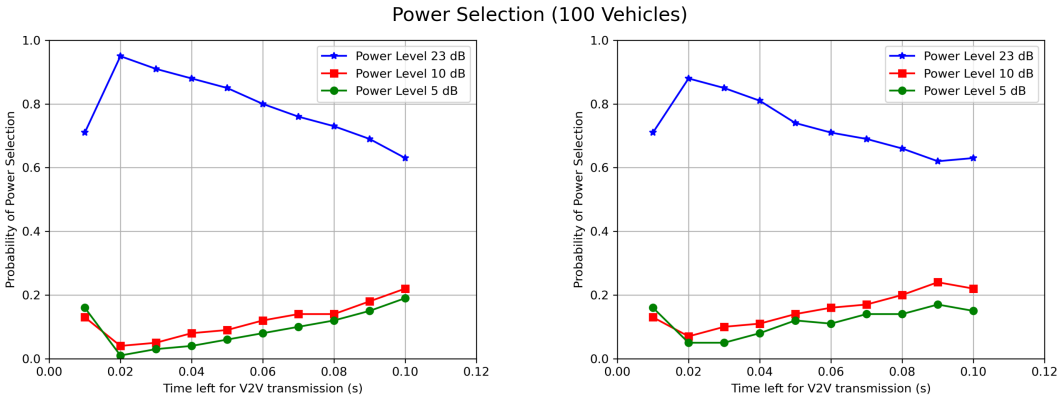

Figure 4.9: Power Selection without (Left) and with (Right) ICM (100 Vehicles)

From this pair of diagrams, we can observe that in double Q learning, the agent has a very high probability to choose maximum power level for transmission, particularly when there are 20 to 40 milliseconds left. At 20ms, the probability almost goes to 100%, whereas the probability of choosing the lowest power level is close to 0%.

With intrinsic reward where the agent is motivated to explore different actions, the probability of choosing the highest power level is significantly lower, although its ability to distinguish between medium and low power levels decrease in dense traffic scenario, the proposed ICM-DQRA method still outperforms double Q learning.

To better summarize the power selection results that ICM brings, Table 4.3 shows the average transmit power selection results of the two algorithm versus number of vehicles.

Table 4.3: Average Power Level (dBm)

No. Vehicles	**20**	**40**	**60**	**80**	**100**
Double Q	19.454	19.593	19.624	19.727	19.853
ICM-DQRA	18.077	18.445	18.622	18.850	18.915

Following equation (4.1), we calculate the average transmit power of a V2V link in milliwatts(mW):

$$P = 10^{\left(\frac{P_0}{10}\right)} \tag{4.1}$$

where P_0 is the average power level.

The the average power consumption of an individual V2V link in shown in figure 4.10

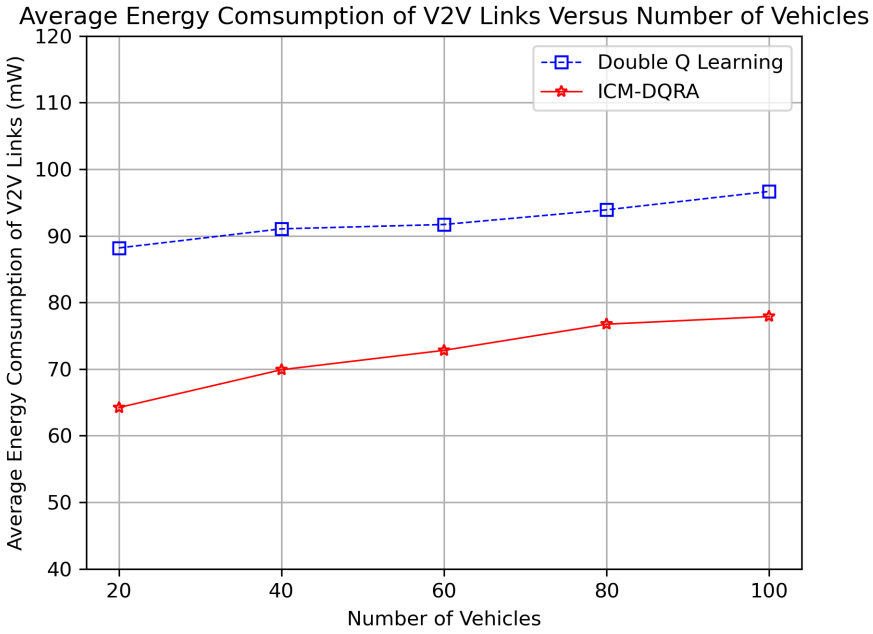

Figure 4.10: Average Energy Consumption of V2V Links

It is observed that a decrease in energy consumption is achieved by the ICM-DQRA algorithm during transmission. The decrease level ranges from roughly 20% in dense traffic scenario to 27% in sparse traffic scenario. This improvement follows naturally from the agent's power selection decisions.

To conclude, the proposed algorithm is able to decrease the energy consumption by dynamically adjusting the agents' resource assignment while at the same time, satisfy the latency constraint and maintain high V2I capacities.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Over the past years, reinforcement learning algorithms have been employed to resource allocation problems in vehicular networks. In this thesis, we propose the ICM-DQRA method to provide an efficient way for resource allocation. The main objective is to reduce the energy consumption while satisfying the latency constraints.

To achieve this, and to handle potential drawbacks that traditional reinforcement learning algorithms face, such as scalability issue and exploration challenge to the agent, we introduced an ICM-based framework to capture a meaningful feature representation from the agent's state representation, and introduce an intrinsic reward to double Q learning. The proposed ICM-DQRA enables the agent to explore a larger action space, and solves the exploration challenge the agent might face. In addition, we studied the resource allocation problem in both 4G and 5G settings, and compared the failure rate and V2I capacity with a greedy DQN algorithm.

Our results indicate that both double Q learning and ICM-DQRA have higher V2I capacity and lower failure rate than the greedy solution. The main difference between 4G and 5G settings lie in the failure rate in dense traffic scenario. With intrinsic reward, the agents will choose power levels for transmission in a less aggressive way than double Q learning. Though this finding is promising with respect to reducing energy consumption and overcome the bottleneck 5G and beyond research faces, we also noticed that the percentage of energy reduction decreases by up to 7% in dense traffic scenario.

5.2 Future Work

The outcomes of this thesis provides several directions for future research:

- Variants to The ICM Agent: The ICM agent consists of three parts with the

feature extractor model extracting feature representations of states. In this thesis we used RNN architecture for this model. However, there might be other variants to this architecture that might lead to better performance in dense traffic scenario. Specifically, we can extract the time-series information from the agent's state information. Table 5.1 shows the structure of replay memory.

Assume we have 60 V2V links in the communication system, the replay memory can be split into groups of 60 entries. The first 60 entries stores the state information of the 60 links at time step t=0. They are followed by another group of 60 entries with state information for all links at time step $t+1$. With this we will be able to either pre-process the memory, or provide additional inputs to the ICM agent to capture the time-series information in the states. This will give us a wider range of machine learning models to choose from for the feature extractor model.

Table 5.1: Replay Memory Structure

Entry	Content
0	$(s_t^1, a_t^1, r_t^1, s_{t+1}^1)$
1	$(s_t^2, a_t^2, r_t^2, s_{t+1}^2)$
...	...
59	$(s_t^{60}, a_t^{60}, r_t^{60}, s_{t+1}^{60})$
60	$(s_{t+1}^1, a_{t+1}^1, r_{t+1}^1, s_{t+2}^1)$
...	...
119	$(s_{t+1}^{60}, a_{t+1}^{60}, r_{t+1}^{60}, s_{t+2}^{60})$
120	$(s_{t+2}^1, a_{t+2}^1, r_{t+2}^1, s_{t+3}^1)$
...	...

- Exploration of Multi-agent Learning: Since each V2V link in the communication network operates independently, we treat the whole system as single-agent from the perspective of reinforcement learning. It's worthwhile to dive into the realm of multi-agent reinforcement learning to see what intrinsic reward can bring.

- Exploration of more state-of-the-art algorithms: The focus of this thesis has been to configure ICM into double Q learning. We focused the Q-learning family algorithms and mainly tested the performance of double Q-learning with and without intrinsic reward. Though in theory, the ICM agent has the flexibility to

be built onto any existing reinforcement learning algorithms [18]. Therefore it is worthwhile to attempt to configure ICM into more state-of-the-art algorithms, such as Proximal Policy Optimization (PPO) algorithm and its variants to see what intrinsic reward can bring.

These prospects for future work illustrate the vast potential for further exploration in this field, potentially leading to more sophisticated and promising results.

Bibliography

[1] 3rd Generation Partnership Project (3GPP). Study on New Radio Access Technology - Radio Access Architecture and Interfaces. Technical Report 36.885, 3GPP, June 2016. Version 2.0.0, Release 14.

[2] Fakhar Abbas, Pingzhi Fan, and Zahid Khan. A novel low-latency V2V resource allocation scheme based on cellular V2X communications. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2185–2197, 2019.

[3] Mariem Allouch, Sondes Khemiri-Kallel, Ahmed Soua, and Samir Tohme. A priority and guarantee-based resource allocation with reuse mechanism in LTE-V mode 3. In *2021 Wireless Days (WD)*, pages 1–5, 2021.

[4] Fadi AlMahamid and Katarina Grolinger. Reinforcement learning algorithms: An overview and classification. In *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–7, 2021.

[5] Yuanfeng Ding, Yan Huang, Li Tang, Xizhong Qin, and Zhenhong Jia. Resource allocation in V2X communications based on multi-agent reinforcement learning with attention mechanism. *Mathematics*, 10(19):3415, 2022.

[6] Federal Communications Commission. Use of the 5.850-5.925 GHz Band. Proposed Rule, May 2021. Federal Register.

[7] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Leibo, and Audrunas Gruslys. Deep Q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, 04 2018.

[8] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, 7:133653–133667, 2019.

[9] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *CoRR*, cs.AI/9605103, 1996.

[10] Sagar Kavaiya. Learn with curiosity: A hybrid reinforcement learning approach for resource allocation for 6G enabled connected cars. *Mobile Networks and Applications*, pages 1–11, 06 2023.

[11] Hyun-Suk Lee, Jin-Young Kim, and Jang-Won Lee. Resource allocation in wireless networks with deep reinforcement learning: A circumstance-independent approach. *IEEE Systems Journal*, 14(2):2589–2592, 2020.

[12] Bosen Li, Dazhi He, Yijia Feng, Yin Xu, and Hongjiang Zheng. Spectrum resource allocation scheme for alarm information delivery in V2V communication. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, 2018.

[13] Shoufeng Lu, Ximin Liu, and Shiqiang Dai. Q-learning for adaptive traffic signal control based on delay minimization strategy. pages 687–691, 04 2008.

[14] David Matolak. V2V communication channels: State of knowledge, new results, and what's next. pages 1–21, 05 2013.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[16] Shahid Mumtaz, Anwer Al-Dulaimi, Valerio Frascolla, Dusit Niyato, and Keith Briggs. Dynamic spectrum management for 5G. *IEEE Wireless Communications*, 24(5):12–13, 2017.

[17] Silvia Mura, Francesco Linsalata, Marouan Mizmizi, Maurizio Magarini, Majid Nasiri Khormuji, Peng Wang, Alberto Perotti, and Umberto Spagnolini. Spatial-interference aware cooperative resource allocation for 5G V2V communications. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pages 1–6, 2022.

[18] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017.

[19] Chandrashekhar S Pawar and Rajnikant B Wagh. Priority based dynamic resource allocation in cloud computing. In *2012 International Symposium on Cloud and Services Computing*, pages 1–6. IEEE, 2012.

[20] Guannan Qu, Adam Wierman, and Na Li. Scalable reinforcement learning for multiagent networked systems. *Operations Research*, 70(6):3601–3628, 2022.

[21] Zhenjiang Shi and Jiajia Liu. Sparse code multiple access assisted resource allocation for 5G V2X communications. *IEEE Transactions on Communications*, 70:1–1, 10 2022.

[22] Brahmjit Singh and Sandeepika Sharma. Enhanced autonomous resource selection algorithm for cooperative awareness in vehicular communication. In *2019 International Conference on High Performance Computing & Simulation (HPCS)*, pages 324–328, 2019.

[23] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning, 2015.

[24] Shujie Wu, Qi Yang, and Xuemin Hong. Joint spectrum resource allocation and power control for LTE-V2V communication. In *2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 44–48, 2022.

[25] Hao Ye and Geoffrey Ye Li. Deep reinforcement learning for resource allocation in V2V communications. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[26] Wonsuk Yoo, Jusik Yun, Jaewook Jung, Giyoung Hwang, and Jong-Moon Chung. Optimized resource utilization scheme for real-time V2V sidelink unicast communication in 5G networks. *IEEE Wireless Communications Letters*, 12(10):1721–1725, 2023.

[27] Shunliang Zhang. An overview of network slicing for 5g. *IEEE Wireless Communications*, 26(3):111–117, 2019.