Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065860

ARTICLE

Resource Allocation in V2X Networks: A Double Deep Q-Network Approach with Graph Neural Networks

Zhengda Huan[1], Jian Sun[2,*], Zeyu Chen[1], Ziyi Zhang[1], Xiao Sun[1] and Zenghui Xiao[1]

[1]School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China
[2]School of Electrical and Electronics Engineering, Shandong University of Technology, Zibo, 255000, China

*Corresponding Author: Jian Sun. Email: sj103063@163.com

Received: 23 March 2025; Accepted: 12 June 2025; Published: 30 July 2025

ABSTRACT: With the advancement of Vehicle-to-Everything (V2X) technology, efficient resource allocation in dynamic vehicular networks has become a critical challenge for achieving optimal performance. Existing methods suffer from high computational complexity and decision latency under high-density traffic and heterogeneous network conditions. To address these challenges, this study presents an innovative framework that combines Graph Neural Networks (GNNs) with a Double Deep Q-Network (DDQN), utilizing dynamic graph structures and reinforcement learning. An adaptive neighbor sampling mechanism is introduced to dynamically select the most relevant neighbors based on interference levels and network topology, thereby improving decision accuracy and efficiency. Meanwhile, the framework models communication links as nodes and interference relationships as edges, effectively capturing the direct impact of interference on resource allocation while reducing computational complexity and preserving critical interaction information. Employing an aggregation mechanism based on the Graph Attention Network (GAT), it dynamically adjusts the neighbor sampling scope and performs attention-weighted aggregation based on node importance, ensuring more efficient and adaptive resource management. This design ensures reliable Vehicle-to-Vehicle (V2V) communication while maintaining high Vehicle-to-Infrastructure (V2I) throughput. The framework retains the global feature learning capabilities of GNNs and supports distributed network deployment, allowing vehicles to extract low-dimensional graph embeddings from local observations for real-time resource decisions. Experimental results demonstrate that the proposed method significantly reduces computational overhead, mitigates latency, and improves resource utilization efficiency in vehicular networks under complex traffic scenarios. This research not only provides a novel solution to resource allocation challenges in V2X networks but also advances the application of DDQN in intelligent transportation systems, offering substantial theoretical significance and practical value.

KEYWORDS: Resource allocation; V2X; double deep Q-network; graph neural network

1 Introduction

In the context of smart cities, V2X plays a crucial role in intelligent transportation systems. This technology enables comprehensive communication between vehicles and their surrounding environments, including V2V, V2I, Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) communication [1]. As the automotive industry advances towards technologies such as autonomous driving, intelligent navigation, and automated parking, the importance of V2X continues to grow. However, V2X still faces critical challenges in balancing communication performance and safety constraints.

To address these issues, various V2X technologies have been developed. Recently, among these technologies, Cellular Vehicle-to-Everything (C-V2X) has been recognized as a critical technology [2]. It offers higher data transmission rates, lower latency, and better reliability than IEEE 802.11p technology. Additionally, the 3rd Generation Partnership Project (3GPP) has standardized New Radio Vehicle-to-Everything (NR-V2X) technology in its Release 16 standards [3]. However, with the deployment of these advanced technologies, new challenges arise. One key challenge is the resource allocation problem, which is crucial to support the substantial wireless communication demands of V2X. This problem is NP-hard, making it difficult to simultaneously satisfy the reliability requirements of V2V links and the rate requirements of V2I links in vehicular networks. Traditional resource allocation methods rely on accurate Channel State Information (CSI), which is difficult to obtain in high-speed vehicular environments.

These new challenges are difficult to address with traditional methods, prompting research to shift towards advanced algorithms [4]. In recent years, deep learning methods have developed rapidly, particularly Deep Reinforcement Learning (DRL), which leverages the powerful function approximation capabilities of deep learning. Reference [5] addresses the spectrum sharing problem in vehicular networks using multi-agent reinforcement learning, where multiple V2V links share a frequency spectrum with V2I links. DRL is particularly effective for solving resource allocation problems, especially in environments where CSI is unreliable. By continuously learning through trial and error to develop strategies that maximize long-term rewards, DRL ultimately enhances the performance of distributed resource allocation systems. DRL has also been explored as a tool for enhancing energy efficiency in vehicular communications. For instance, references [6] and [7] demonstrate how reinforcement learning can minimize transmission power or reduce energy consumption while maintaining communication reliability, highlighting the growing importance of energy-aware decision-making in V2X resource allocation.

To address these challenges, researchers have explored various machine learning methods, such as CNN, DNN, LSTM, and GNN [8]. Among them, GNN has demonstrated exceptional performance. Reference [9] proposes an edge-update mechanism for GNNs to efficiently manage radio resources in wireless networks. This approach improves the sum rate and reduces computation time compared to state-of-the-art methods while demonstrating strong scalability and generalization.

Although recent studies have applied GNNs to resource allocation in vehicular networks, such efforts remain limited in scope and number [10]. These approaches often model the network as a complete graph, assuming that all communication links mutually influence each other. While such modeling can be effective under moderate traffic conditions, it becomes computationally infeasible in high-density scenarios, resulting in excessive complexity and unacceptable decision latency for real-time applications. Furthermore, most existing GNN-based methods rely on fixed or predefined neighbor sampling strategies, which are ill-suited to the dynamic and heterogeneous interference patterns characteristic of vehicular environments. This limitation often leads to suboptimal resource allocation outcomes.

In addition, few GNN-based solutions consider the trade-off between communication performance and energy efficiency—an increasingly important issue as vehicular networks move toward electric and autonomous systems. Recent studies [11] have shown that graph-based representations can support not only interference modeling but also energy-efficient decision-making in large-scale networks. Parallel to these challenges, traditional DQN-based reinforcement learning frameworks suffer from overestimation bias due to the shared use of a single network for both action selection and evaluation. This issue compromises training stability and weakens policy robustness in complex, dynamic environments [12]. In addition to communication and energy-related constraints, resource allocation in vehicular networks must also address task-level requirements, particularly in multi-agent systems. For instance, integrated task assignment and path planning under capacity constraints, such as those studied in capacitated multi-agent pickup and delivery problems, are highly relevant to cooperative vehicular environments. These problems involve combinatorial complexity and inter-agent dependencies, directly influencing how communication and computational resources should be dynamically distributed.

To overcome the aforementioned limitations, this paper proposes an innovative framework that integrates GNNs and DRL to enhance the efficiency of distributed resource allocation and mitigate the impact of inaccurate local observations. Specifically, the framework constructs a dynamic graph with communication links as nodes and inter-link interference as edges, enabling adaptive adjustments to the network topology. It incorporates three key innovations: adaptive graph construction with dynamic neighbor sampling based on interference and spatial relevance, GAT-based personalized feature aggregation, and the integration of GNN with Double DQN to improve stability and decision quality in dynamic vehicular environments.

2 System Model and Problem Formulation

2.1 Basic Structure

Building on the challenges discussed earlier, this section focuses on modeling the V2I and V2V communication scenarios in the V2X paradigm, particularly targeting efficient resource allocation in dynamic vehicular networks. As shown in Fig. 1, the system is modeled based on a vehicular traffic scenario at an intersection, where the Base Station (BS) is located at the center. Vehicles enter the roads with randomly selected speeds and maintain a constant velocity. V2V communication is primarily used for exchanging safety-critical messages between vehicles, while V2I links support high-throughput data transmission to the BS, such as infotainment content. The resource allocation mechanism follows C-V2X mode-4, where vehicles, as agents in the system, use DDQN to determine the best subchannel and power level [13]. A shared reward function is designed to jointly optimize the selection of spectrum and power levels in the network. This decentralized resource allocation framework leverages local vehicle interactions and learned policies to ensure efficient and effective resource utilization in dynamic vehicular environments.

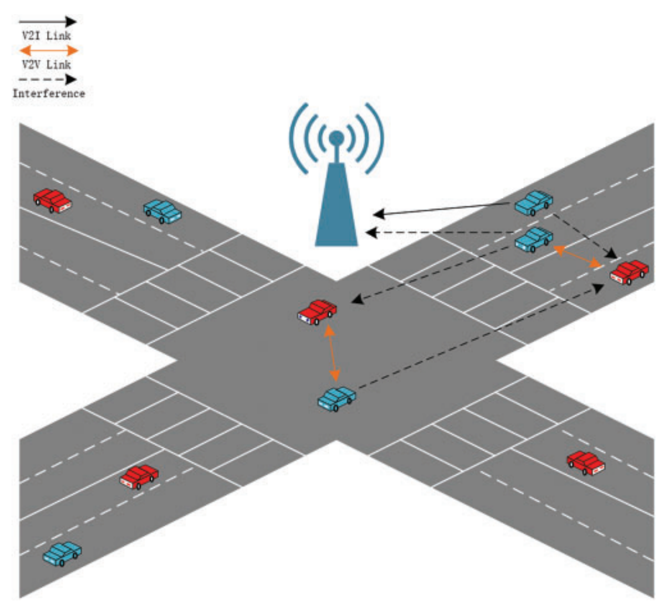

Figure 1: System model

In this model, resource allocation is divided into channel selection and power level selection. The efficient use of channel resources is key to maximizing system efficiency, as congestion on some channels can lead to underutilization of others. Vehicles, when selecting channels, must consider not only their own communication needs but also the potential impact on other vehicles, especially in cases where the available channel is limited. Power level selection follows channel selection and is crucial for balancing the trade-off between minimizing interference and ensuring reliable communication. Lower transmission power reduces interference but can cause signal failure, whereas higher power increases interference and energy consumption.

The system employs a decentralized approach for V2X resource allocation, assuming that V2I resources are allocated by the BS, and the number of subchannels and power levels is predefined [14]. Vehicles make their own decisions regarding resource allocation, with each V2V link having multiple possible resource choices. The goal is to minimize interference to V2I links while meeting latency and reliability requirements for V2V communication, thereby optimizing the overall system performance [15].

To achieve this, the interaction between the model and the environment occurs at two time scales: a larger time scale for determining neighbor relationships and a smaller time scale for gathering local observations and neighbor data. The vehicles process this information through a GNN model to generate low-dimensional feature vectors representing global information. These vectors are then used by DRL to make decisions about channel and power level selections.

2.2 Interference Calculation Method

We assume there are m cellular users (CUEs) and k V2V user pairs (VUEs), denoted by sets $\mathcal{M} = \{1, 2, \ldots, m\}$ and $\mathcal{K} = \{1, 2, \ldots, k\}$, respectively. V2V links reuse the uplink spectrum that is orthogonally allocated to V2I links.

The Signal-to-Interference-plus-Noise Ratio (SINR) for the i-th CUE and j-th VUE is calculated as follows:

$$\gamma^c[i] = \frac{P_i^c \, h_i}{\sigma^2 \; + \; \sum_{j \in K} \rho_j[i] \, P_j^v \tilde{h}_j} \tag{1}$$

$$\gamma^v[j] = \frac{p_j^v \cdot g_j}{\sigma^2 + \sum_{i \in M} \rho_j[i] p_i^c g_{i,j} + G_{V2V}} \tag{2}$$

$$G_{V2V} = \sum_{i \in M} \rho_j[i] \sum_{j' \in \mathcal{K},\, j' \neq j} \rho_{j'}[i] p_{j'}^v g_{j'j}^v \tag{3}$$

Based on Shannon's capacity formula, the achievable communication rates for the i-th CUE and the j-th VUE are given by:

$$C^c[i] = B \cdot \log(1 + \gamma^c[i]), \quad C^v[j] = B \cdot \log(1 + \gamma^v[j]) \tag{4}$$

All related symbols used in the above equations are summarized in Table 1.

Table 1: Symbol definitions in SINR and capacity calculations

Symbol	Description
P_i^c	Transmit power of the i-th CUE
P_j^v	Transmit power of the j-th VUE

(Continued)

Table 1 (continued)

Symbol	Description
h_i	Uplink channel gain from CUE i to the BS
$\tilde{h}_j$	Uplink channel gain from VUE j to the BS
g_j	Direct channel gain of the j-th VUE link (VUE-Tx to VUE-Rx)
$g_{i,j}$	Interference channel gain from CUE i to VUE j
$g^v_{j',j}$	Interference channel gain from VUE j' to VUE j
$\rho_j[i]$	Indicator: 1 if VUE j shares spectrum with CUE i; 0 otherwise
σ^2	Noise power
B	Channel bandwidth
G_{V2V}	Total interference from other VUEs on the same subchannel

Note: All gain values follow path-loss and shadowing models consistent with 3GPP TR 36.885 channel modeling assumptions.

3 Graph Neural Network Models

3.1 Neighbor Sampling and Graph Construction

In V2X networks, the dynamic nature of vehicle movement and interference makes efficient graph construction essential for real-time resource allocation. Traditional approaches often use fixed neighbor sampling or model the network as a complete graph, which results in excessive complexity and decision latency in high-density environments.

To address this, we propose an adaptive neighbor sampling method that evaluates the importance of each neighbor based on two factors: interference level and distance to the BS. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_S\}$ denote the set of vehicles. For each vehicle v_i, the importance of a neighboring vehicle v_j is calculated as:

$$I_j = w_1 \cdot f_{\mathrm{int}}(\gamma_{v,ij}) + w_2 \cdot f_{\mathrm{dist}}(d_j) \tag{5}$$

where $\gamma_{v,ij}$ denotes the estimated interference level between vehicles v_i and v_j, while d_j represents the Euclidean distance from v_j to the BS. The functions $f_{\mathrm{int}}(\cdot)$ and $f_{\mathrm{dist}}(\cdot)$ are normalization functions that scale their respective inputs to the range $[0,1]$. The weights w_1 and w_2 respectively represent the relative importance of interference and spatial proximity, subject to the constraint $w_1 + w_2 = 1$.

Each node selects the top five neighbors based on this score. A graph is then constructed where V2V links are nodes and interference relationships form the edges [16]. We apply GAT to aggregate node features efficiently. This adaptive approach limits graph size, preserves critical interference relationships, and supports scalable learning in dense vehicular scenarios.

3.2 GAT

Since we assumed there are k pairs of VUEs, the nodes in the graph can be represented as $P = [p_1, p_2, \ldots, p_k]$. For node p, it contains a list storing the indices of its neighboring nodes as well as an initial feature vector x_p. The vector x_p encapsulates the local observations of the vehicle's channel and interference information. We assume that the number of subchannels is equal to the number of CUEs, denoted as m. On the i-th subchannel, the instantaneous channel power gain of the V2V link is $G_t[i]$, the subchannel power gain of the V2V link is $H_t[i]$, and the interference signal strength for the one-time slot is

$I_{t-1}[i]$. Therefore, the feature of node p can be expressed as:

$$x_p = \{G_t \| H_t \| I_{t-1}\} \tag{6}$$

The aggregation function of the GAT introduces a self-attention mechanism, assigning a weight to each neighboring node. This weight reflects the influence of the neighbor on the information propagation to the target node; the larger the weight, the stronger the influence of the neighbor. Unlike conventional graph convolution methods, GAT enables each node to dynamically compute attention weights based on neighbor features, thereby facilitating structure-aware and adaptive feature aggregation.

For each node $p_i \in P$, the influence of neighboring nodes $p_j \in P$ during information aggregation is determined by calculating attention weights that reflect the feature similarity between p_i and p_j. Assume that N_i contains the node coefficients excluding i. These attention weights are computed via a Feed-Forward Neural Network (FFNN) as follows:

For node p_i and neighbor p_j, their feature vectors are concatenated and transformed through a learnable weight matrix W, followed by a *LeakyReLU* activation function to capture nonlinear relationships. This process is formalized as:

$$e_{ij} = \text{LeakyReLU}\left(W[x_{pi} \| x_{pj}]\right), j \in N_i \tag{7}$$

where $[x_{pi} \| x_{pj}]$ denotes the feature vectors of x_{pi} and x_{pj}, respectively, and W represents the concatenated feature vector.

The attention coefficient α_{ij} between v_i and v_j is derived by normalizing the transformed features via softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{y \in N_i} \exp(e_{iy})} \tag{8}$$

Here, α_{ij} quantifies the influence of neighbor v_j on v_i during feature aggregation, ensuring that all weights sum to 1 across the neighborhood.

With the attention weight α_{ij} for each neighboring node, we aggregate the features of the neighboring nodes using a weighted sum to generate the new feature representation z_{p_i} for node p_i:

$$z_{p_i} = \text{LeakyReLU}\left(\sum_{j \in N_i} \alpha_{ij} W p_j\right) \tag{9}$$

In this manner, the node p_i updates its features based on the features of its neighbors and their corresponding attention weights. This process can be viewed as the propagation and aggregation of information within the graph, as shown in Algorithm 1.

Algorithm 1: GAT-based feature aggregation

Input: Node features $H = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]$; Neighborhood $\mathcal{N}(i)$ for each node i
Output: Updated node features $H' = [\mathbf{h}'_1, \ldots, \mathbf{h}'_N]$
1 **for** *each node* $i \in \{1, \ldots, N\}$ **do**
2   **for** *each neighbor* $j \in \mathcal{N}(i)$ **do**
3   Concatenate features: $\mathbf{h}_{ij} \leftarrow [\mathbf{h}_i \,\|\, \mathbf{h}_j]$;

(Continued)

Algorithm 1 (continued)

4 Compute attention score: $e_{ij} \leftarrow \text{LeakyReLU}(\mathbf{a}^{\top} \cdot \mathbf{h}_{ij})$;
5 Normalize attention scores: $\alpha_{ij} \leftarrow \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})}$;
6 Aggregate features: $\mathbf{h}_i' \leftarrow \sigma\left(\sum_{j \in N(i)} \alpha_{ij} \cdot \mathbf{W}\mathbf{h}_j\right)$;
7 **return** $H' = [\mathbf{h}_1', \ldots, \mathbf{h}_N']$

The GAT enhances node relationship capture by assigning personalized weights to neighboring nodes, effectively modeling complex dependencies in vehicular networks [17]. This approach improves resource allocation accuracy by prioritizing reliable links with good channel quality and high communication success rates, optimizing network performance. GAT adapts to dynamic environmental changes by adjusting weights in real-time, ensuring continuous and effective resource allocation. Furthermore, we adopt the *LeakyReLU* activation function in the GAT aggregation process to address the issue of vanishing gradients and neuron inactivity commonly associated with *ReLU*. *LeakyReLU* allows a small, non-zero gradient for negative input values, which helps maintain learning dynamics and stabilizes the training process. This choice follows the original GAT design and has been shown to improve convergence speed and model robustness in dynamic graph environments [18].

3.3 Real-Time Deployment Considerations of GAT Aggregation

While the GAT enables dynamic and personalized feature aggregation by assigning learnable weights to neighboring nodes, its real-time deployment raises concerns regarding computational complexity, particularly in high-density vehicular networks. In each GAT layer, attention weights are computed for every pair of neighboring nodes, leading to a time complexity of O(E) per layer, where E denotes the number of edges. In scenarios with N nodes and average degree D, this translates to approximately O(N·D) operations. We mitigate the overhead through two mechanisms:

1. Incomplete Graph Construction: By limiting each node to at most 5 neighbors through importance-based sampling, the number of edges per node remains bounded, thereby capping the attention computation cost.

2. Parallelization: The GAT computation is implemented using TensorFlow's parallelized matrix operations, which enables real-time inference within 30 ms per decision round in a simulated environment with 100 vehicles. GAT introduces additional overhead compared to traditional graph convolutions, our design ensures bounded graph size and efficient computation, making it suitable for time-constrained vehicular applications. Since each node maintains a fixed number of neighbors, the total graph complexity increases linearly with the number of vehicles, making the framework structurally scalable to higher-density scenarios without incurring exponential computational growth. Therefore, the proposed approach is also feasible for deployment on modern edge computing devices, which typically have limited computational resources but demand real-time performance.

4 The GNN-DDQN Model For Resource Allocation Problems

DRL combines deep learning and reinforcement learning to enable an agent to learn from its interactions with the environment by optimizing cumulative rewards, treating learning as a heuristic evaluation process [19]. Problems are typically modeled as Markov Decision Processes (MDPs), where the agent observes the state, makes decisions based on a policy, executes actions, and receives rewards that update the state [20]. However, the Deep Q-Network (DQN) suffers from an overestimation bias due to using the same network for both action selection and evaluation, which can degrade learning stability and performance. To mitigate this issue, the DDQN was introduced as an enhanced variant of the original DQN. By decoupling the action selection and value estimation processes, DDQN effectively mitigates the overestimation problem and achieves better performance, particularly in vehicular networks.

Next, we will sequentially introduce the details of the state space, action space and reward in the DDQN network.

State Space: for the V2X environment considered in this study, the true state information primarily includes the vehicle's observations of the environment and the low-dimensional features z_p extracted by the GNN model from these observations. To assist the agent in making better decisions, each vehicle sends its channel selection information to the target vehicle [21]. Accordingly, each agent collects the previous channel selection information N_{t-1} from its neighboring agents and calculate the ratio of the remaining bits to be transmitted to the total bits to be transmitted L_t, as well as the remaining transmission time U_t under the delay constraint. Combining these pieces of information, the state representation obtained by the agent is as follows:

$$S_p^t = \left\{ z_p^t || x_p^t || N_{t-1}^p || L_t^p || U_t^p \right\} \tag{10}$$

Action Space: based on the collected and observed state information, the DDQN network selects an action a_t from the action space A according to the policy π [22]. Since the agent needs to simultaneously choose a sub-channel and a power level, we combine these two types of actions into a composite action. The selected composite action is mapped to two dimensions, corresponding to the choice of sub-channel and power level. In the scenario considered in this paper, we simplify the case by defining n power levels and m resource blocks. Thus, there are a total of nm possible action combinations in the action space. If the agent selects an action a_t, we can decompose it as follows:

$$a_r^t = a_t \% m \tag{11}$$

and

$$a_p^t = a_t / m \tag{12}$$

a_r^t denotes the sub-channel selection action based on the action a_t and the number of resource blocks m, while a_p^t denotes the power level selection action. The specific forms of these two functions are designed according to the characteristics of the DDQN model and the practical requirements of the V2X communication resource allocation problem. The goal is to ensure that the agent can make precise decisions within the action space, thereby achieving efficient resource allocation that meets the performance requirements of V2V and V2I links [23]. This includes ensuring low latency and high reliability for V2V communication while maximizing the transmission rate of the V2I link.

Reward Function: in the reinforcement learning framework, designing an appropriate reward function is crucial for guiding the agent toward optimal decision-making. In the context of V2X communication resource allocation, our objective is to maximize the V2V link's ability to meet low latency and high reliability communication requirements while minimizing interference to the V2I link in order to maximize the V2I link's transmission rate [24]. The reward function is expressed as follows:

$$r_t = \lambda_c \sum_{i \in M} C^c[i] + (1 - \lambda_c) \sum_{j \in K} C^v[j] - \lambda_p (T_0 - U_t) \tag{13}$$

where r_t represents the immediate reward obtained by the agent at time step t, λ_c represents the weight of the V2I link. U_t represents the remaining time, while T_0 denotes the transmission delay limit. Thus, $(T_0 - U_t)$

indicates the elapsed transmission time. Based on the immediate reward formula, the long-term discounted return can be expressed as:

$$R_t = E\left[\sum_{n=0}^{\infty} \beta^n r_{t+n}\right] \tag{14}$$

where $\beta \in [0,1]$ denotes the discount factor for the reward. A higher value of β suggests that the agent prioritizes long-term returns, whereas a lower value indicates a preference for immediate rewards.

Fig. 2 illustrates the structure of the GNN-DDQN framework, which integrates deep reinforcement learning with graph neural networks for resource allocation in vehicular networks. The overall architecture comprises two main modules: the DDQN and the GNN. The DDQN model utilizes a three-layer neural network with 500, 250, and 120 neurons in each layer, respectively. This network structure was chosen based on empirical tuning to balance model complexity and training performance. The initial learning rate is 0.01, and a β-greedy exploration strategy is employed. The DDQN framework includes agents interacting with the vehicular environment, where at each time step t, an agent observes the current state s_t and selects an action a_t from the action space A based on its policy π [25]. This action, typically involving the selection of a transmission channel and power level for the V2V link, is guided by the state-action value function $Q(s_t, a_t)$, which is approximated using deep neural networks to adapt to complex environmental dynamics. Following the action execution, the environment transitions to a new state s_{t+1} and provides a reward r_t that is derived from the interruption probability of the V2V link and the throughput of the V2I link, as shown in Algorithm 2.

Algorithm 2: Training procedure of GNN-DDQN

Input: Replay buffer $\mathcal{D}$, discount factor γ, learning rate α, update interval C
Output: Trained Q-network Q_θ

```
Initialize Q-network Q_θ and target network Q_θ-;
Initialize replay buffer D;
for episode = 1 to M do
    Reset environment and get initial state s_0;
    for t = 1 to T do
        Construct communication graph and extract node features using GAT;
        Select action a_t via ε-greedy strategy based on Q_θ(s_t, a);
        Execute a_t, observe reward r_t and next state s_{t+1};
        Store (s_t, a_t, r_t, s_{t+1}) into D;
        Sample mini-batch from D;
        foreach sample (s, a, r, s') in batch do
            Compute target: y ← r + γ · max_{a'} Q_θ-(s', a');
            Update Q_θ by minimizing loss: L ← (y − Q_θ(s, a))^2;
        if t mod C = 0 then
            Update target network: Q_θ- ← Q_θ;
return Q_θ
```

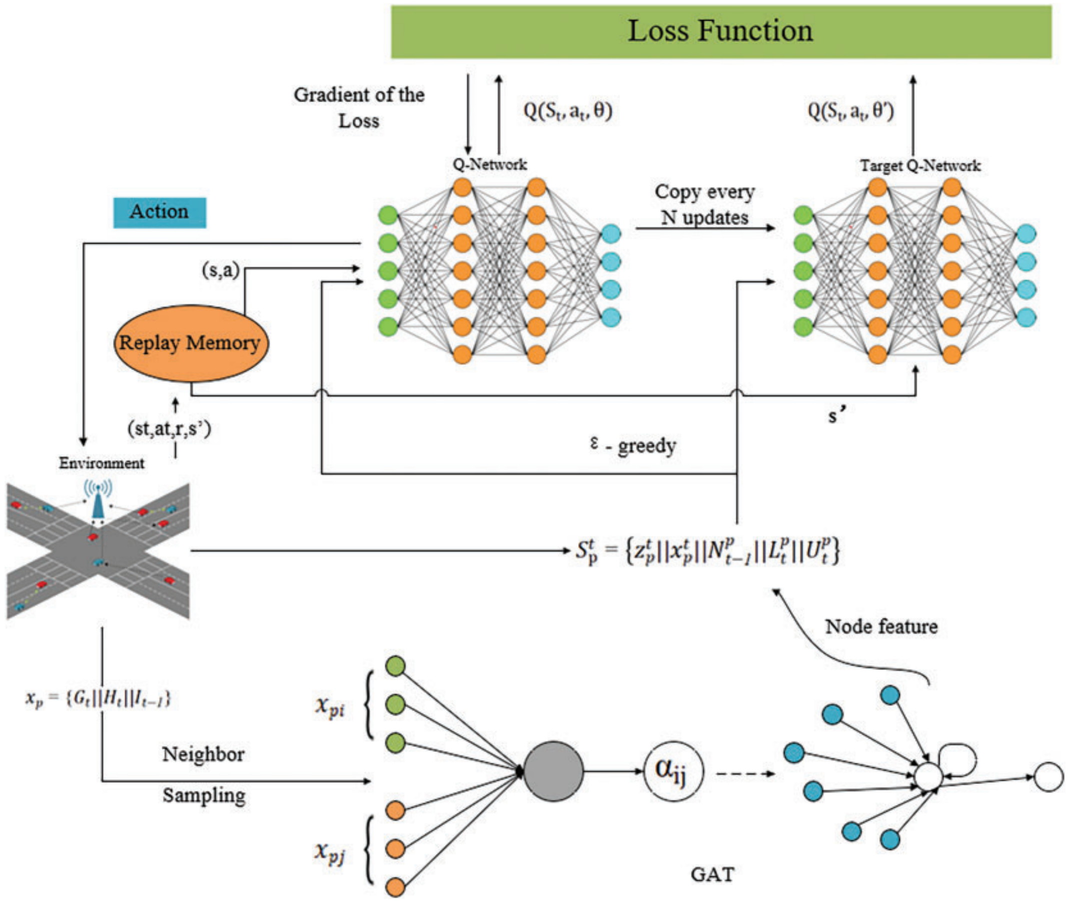

Figure 2: The structure of GNN-DDQN

The GNN module operates in two steps: constructing a graph that captures the global network topology and extracting low-dimensional feature embeddings that represent global information [26]. These embeddings, combined with local observations such as channel state and interference data, enhance the agent's decision-making process. A replay memory module stores experience tuples (s, a, r, s') generated during interactions, which are subsequently used to train the network. At the core of the framework, the Q-Network approximates $Q(s, a, \theta)$ to guide action selection, while a periodically updated Target-Network provides stable target Q-values. Additionally, a GAT aggregates node features using attention coefficients computed from local observations, resulting in low-dimensional embeddings that effectively capture the global context.

This integrated structure, which combines global context awareness with temporal decision-making, enables adaptive and efficient resource allocation in high-density vehicular networks.

5 Simulations

In this section, we present simulation setup and simulation results to show the performance of the proposed GNN-DDQN based resource allocation framework in terms of computational complexity, latency, and resource utilization efficiency, and compare it with other methods.

In this study, the code is configured using Python 3.6.13 and TensorFlow 2.3.1. We consider a single-cell system with a carrier frequency of 2 GHz. The simulation follows the Manhattan scenario setup described in

3GPP TR 36.885 [27], which includes 9 blocks and employs both line-of-sight (LOS) and non-line-of-sight (NLOS) channels. More detailed parameter settings are provided in Table 2. The parameters such as "Carrier frequency", "Bandwidth of single subchannel", "Height of BS Antenna", and others are adapted from [28].

Table 2: System parameters

Description	Specification
Carrier frequency	2 GHz
Bandwidth of single subchannel	1.5 MHz
Height of BS Antenna	25 m
Gain of BS antenna	8 dBi
Noise figure of BS receiver	5 dB
Gain of vehicle antenna	3 dBi
Vehicle speed	36 to 54 km/h
Neighbor vehicles distance threshold	150 m
Number of lanes	4 per direction, total 16 lanes
Noise power	–114 dBm
Maximum delay for V2V link	100 ms
V2V transmission power level list	[23, 10, 5] dBm
$[\lambda_c, \lambda_p]$	[0.3, 1]

Vehicle positions are initialized according to a spatial Poisson process, and their movements follow random waypoint mobility across a 9-block urban grid. Speeds are randomly chosen within the range of 36–54 km/h, consistent with typical city driving. Although no real traffic datasets were used directly, we validated the simulation parameters by comparing with characteristics from benchmark traffic datasets such as TAPAS Cologne and the Luxembourg SUMO dataset. These comparisons ensured that our synthetic setup reflects realistic vehicle density, spacing, and mobility patterns commonly observed in urban environments. The modular design of the proposed GNN-DDQN framework enables it to adapt to varying urban layouts and traffic conditions. Experimental settings can be adjusted to simulate different topologies and mobility models, indicating the model's potential generalization capability across diverse city scenarios.

Building on this simulation environment, we implement the proposed GNN-DDQN model. A GAT with two layers is used to extract structural features, where each node selects up to five neighbors based on an adaptive importance score. The node input feature is 60-dimensional, including channel gain, subchannel gain, and past interference. The GAT outputs a 20-dimensional embedding through attention-weighted aggregation. This embedding, combined with auxiliary features such as remaining transmission time and neighbor activity, forms a 102-dimensional input to a three-layer DDQN network. The network outputs Q-values for 60 discrete subchannel-power actions using the activation function.

Fig. 3 shows how the number of selected neighbors affects the average scaled reward in the GNN-DDQN framework. As the number increases from 1 to 5, the reward improves steadily and peaks at 5 neighbors, suggesting that aggregating information from a moderate set of nearby nodes helps enhance decision-making. Beyond this point, performance declines, likely due to added noise from less relevant neighbors. This highlights the need to choose a suitable neighbor count that balances useful information and noise in dynamic vehicular environments.

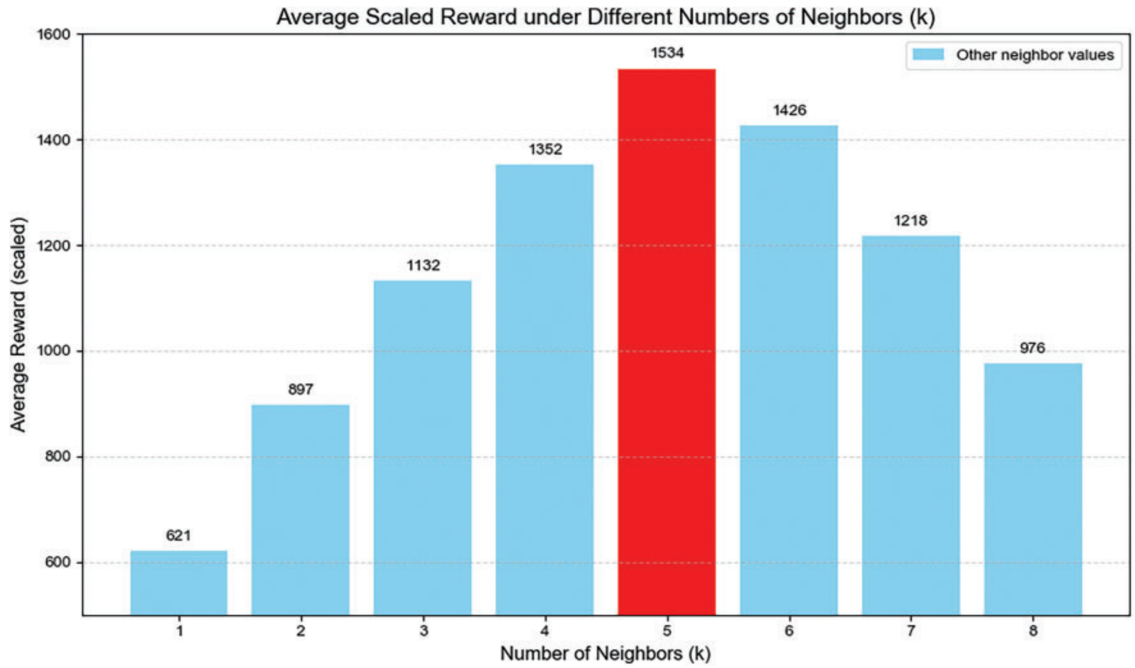

Figure 3: Decision time comparison between complete graph and incomplete graph

Fig. 4 demonstrates the performance of the GNN-DDQN network at different training iterations in a simulated environment. It can be seen that as the number of training iterations increases, the V2I communication rate and the average V2V communication success rate gradually improve and eventually converge. This validates that the model's performance is continuously optimized during the training process, reflecting the positive impact of training on enhancing communication effectiveness.

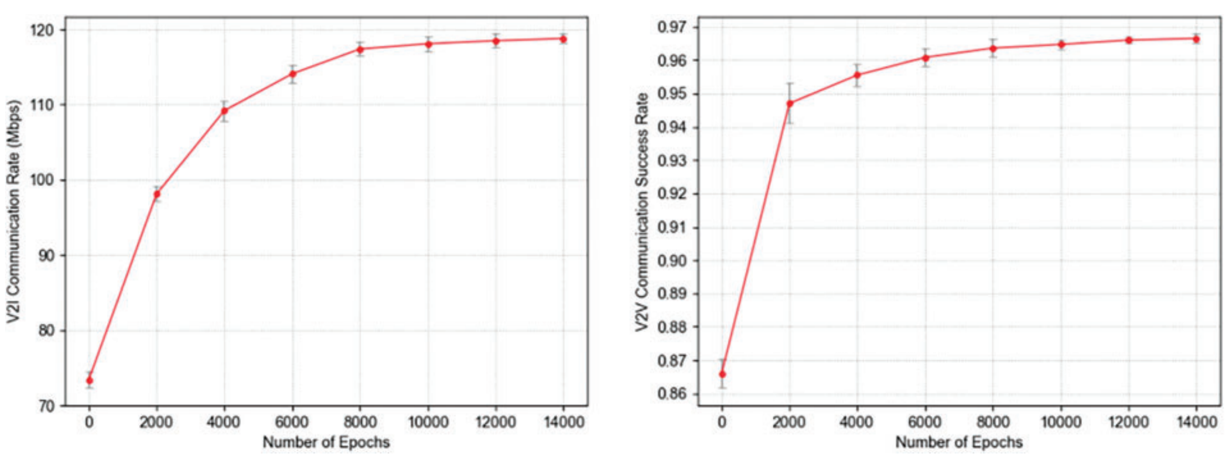

Figure 4: Training effect of V2I communication rate and training effect of V2V communication success rate

Fig. 5 illustrates the impact of increasing vehicle numbers on the average V2I throughput and V2V communication success rate under different resource allocation strategies. As the number of participating vehicles increases, all schemes exhibit a decline in V2I throughput due to intensified interference from V2V links. Simultaneously, V2V communication reliability also decreases as a result of growing channel contention and congestion. Despite these challenges, the proposed GNN-DDQN method significantly outperforms other approaches in both metrics. Across all vehicle densities, GNN-DDQN improves V2I throughput by 22.1% over DQN and by 149.6% over the random baseline. In the high-density scenario with 120 vehicles, it outperforms the four benchmark methods by 310.43%, 36.37%, 34.97%, and 8.81%, respectively. Regarding V2V communication, GNN-DDQN consistently maintains a higher success rate across all densities, with improvements of up to 0.96% over DQN and 9.78% over the random scheme. Under high-density conditions, it achieves V2V reliability gains of 19.53%, 2.08%, 1.91%, and 1.20% compared to the four baselines. These results demonstrate the effectiveness of graph-based structural modeling and attention-guided policy learning in enhancing both V2I and V2V communication performance in dense vehicular networks.

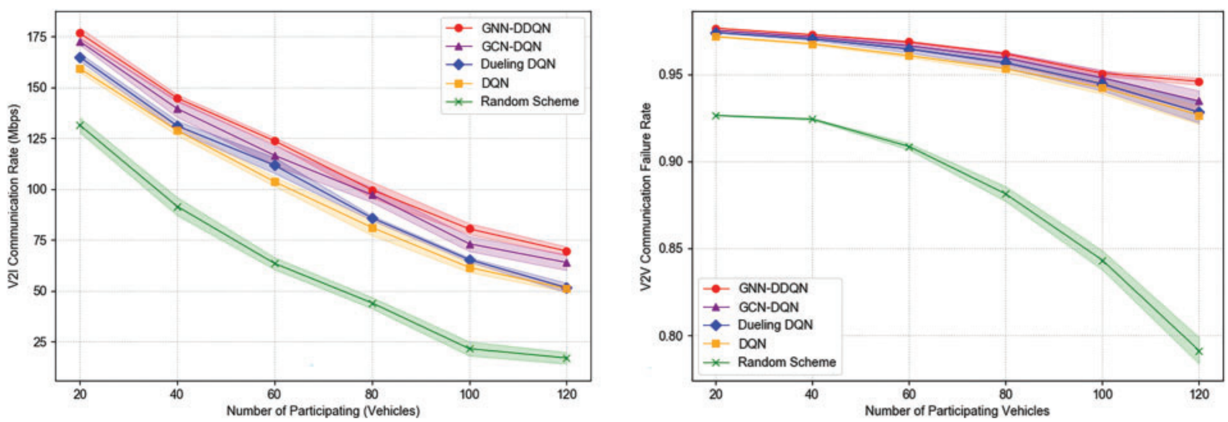

Figure 5: Relationship between number of vehicles and communication performance

Fig. 6 compares the decision-making time when constructing the GNN using a complete graph vs. an incomplete graph, across varying numbers of participating vehicles. As the number of vehicles increases, the decision time under the complete graph configuration rises significantly—from 0.0054 s at 20 vehicles to 0.0078 s at 120 vehicles, representing a 44.44% increase. In contrast, the incomplete graph maintains a relatively stable decision time, fluctuating slightly from 0.0056 to 0.0061 s, with only an 8.93% increase over the same range. At higher vehicle densities, the computational advantage of the incomplete graph becomes more evident. Specifically, at 120 vehicles, the incomplete graph achieves a 21.79% reduction in decision time compared to the complete graph. This stability is primarily attributed to the limited and fixed number of neighbors in the incomplete graph, which avoids the linear growth in computation caused by the increasing number of nodes in the complete graph.

Fig. 7 illustrates the average reward per epoch during training for five model configurations: DQN, DDQN, GCN-DQN, GAT-DDQN and the proposed GAT-DQN. The results show that GAT-DDQN achieves the highest reward and most stable convergence, demonstrating the effectiveness of combining graph attention mechanisms with the DDQN framework for improved learning performance in dynamic environments.

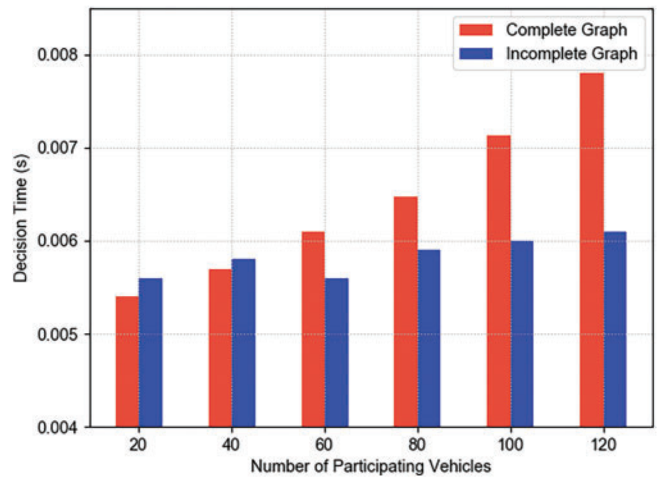

Figure 6: Decision time comparison between complete graph and incomplete graph

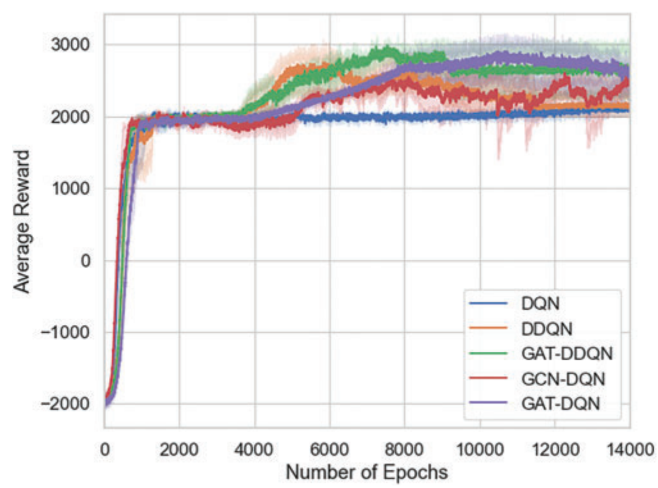

Figure 7: Total reward obtained per epoch during training for different model

6 Conclusions and Future Work

In this paper, we integrate Graph Neural Networks (GNN) with Double DDQN for resource allocation in V2X networks. To curb the exponential growth of computation in dense traffic, we introduce an interference- and location-based neighbor sampling method that limits graph size while retaining critical links. We then build dynamic graphs without additional communication overhead and apply a GAT to weight neighbor features via self-attention. Simulation results demonstrate that GNN-DDQN consistently outperforms standalone DQN. Compared to industrial standards such as Qualcomm's C-V2X solution, our method demonstrates higher flexibility in decentralized V2V communication scenarios and provides better adaptability to dynamic environments due to its learning-based design. Future work will focus on adapting the framework to varying vehicle densities.

While our model addresses channel and power allocation under latency and interference constraints, future work could explore integrated spatio-temporal constraints such as joint vehicle scheduling and trajectory-aware spectrum assignment. Such extensions may benefit from recent progress in multi-agent task routing under capacity and coordination constraints, as investigated in pickup-and-delivery or drone delivery networks. For instance, integrated task assignment and path planning for capacitated multi-agent systems has been shown to offer valuable insights into how spatial-temporal constraints and agent limitations can be jointly optimized [29]. These methodologies could inspire new directions in V2X resource allocation, particularly in scenarios involving mobility prediction, multi-hop relaying, or joint scheduling and routing.

Although the proposed GNN-DDQN framework demonstrates superior performance in simulated high-density vehicular environments, it is important to acknowledge the ethical and safety implications of applying RL in safety-critical V2V communications. The trial-and-error nature of RL may result in suboptimal or unsafe decisions, particularly in early training stages or in highly dynamic, unseen environments. To mitigate these risks, our approach conducts all training in a controlled simulation environment and incorporates latency and reliability constraints directly into the reward function to discourage unsafe behavior. Previous research has shown that properly designed DRL frameworks with safety-aware reward shaping can maintain acceptable QoS levels in vehicular networks [30]. Moreover, the adoption of Graph Neural Networks enhances the model's generalization ability and reduces the chance of unexpected decisions caused by insufficient observations. In future work, we plan to explore safe RL techniques such as Constrained Policy Optimization and hybrid decision architectures to further enhance reliability, interpretability, and regulatory compliance in real-world deployments. We also recognize that the current model has limitations in handling non-stationary environments and scaling to city-level deployments, which will be important directions for our future research.

Acknowledgement: Not applicable.

Funding Statement: Project ZR2023MF111 supported by Shandong Provincial Natural Science Foundation.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, methodology, and writing, Zhengda Huan; supervision and project administration, Jian Sun; validation and data curation, Zeyu Chen and Ziyi Zhang; formal analysis and visualization, Xiao Sun and Zenghui Xiao. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The source code for this study is available in the Gitee repository at https://gitee.com/huan-zhengda/resource-allocation-for-v2-x-communications (accessed on 11 June 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Dhinesh Kumar R, Rammohan A. Revolutionizing intelligent transportation systems with Cellular Vehicle-to-Everything (C-V2X) technology: current trends, use cases, emerging technologies, standardization bodies, industry analytics and future directions. Veh Commun. 2023;43(8):100638. doi:10.1016/j.vehcom.2023.100638.
2. Guo C, Wang C, Cui L, Zhou Q, Li J. Radio resource management for C-V2X: from a hybrid centralized-distributed scheme to a distributed scheme. IEEE J Sel Areas Commun. 2023;41(4):1023–34. doi:10.1109/JSAC.2023.3242723.
3. Bhadauria S, Shabbir Z, Roth-Mandutz E, Fischer G. QoS based deep reinforcement learning for V2X resource allocation. In: 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom); 2020 May 26–2; Online. p. 1–6. doi:10.1109/BlackSeaCom48709.2020.9234960.
4. Liang L, Ye H, Li GY. Spectrum sharing in vehicular networks based on multi-agent reinforcement learning. IEEE J Sel Areas Commun. 2019;37(10):2282–92. doi:10.1109/JSAC.2019.2933962.
5. Zhang Z, Zhang L, Chen Z. Multi-agent reinforcement learning based channel access scheme for underwater optical wireless communication networks. In: 2021 15th International Symposium on Medica0l Information and

Communication Technology (ISMICT); 2021 Apr 14–16; Xiamen, China. p. 65–9. doi:10.1109/ISMICT51748.2021.9434918.

6. Xu Y, Zheng L, Wu X, Tang Y, Liu W, Sun D. Energy-efficient resource allocation for V2X communications. IEEE Internet Things J. 2024;11(18):30014–26. doi:10.1109/JIOT.2024.3410098.
7. Sohaib RM, Onireti O, Sambo Y, Swash R, Imran M. Energy efficient resource allocation framework based on dynamic meta-transfer learning for V2X communications. IEEE Trans Netw Serv Manag. 2024;21(4):4343–56. doi:10.1109/TNSM.2024.3400605.
8. Balkus SV, Wang H, Cornet BD, Mahabal C, Ngo H, Fang H. A survey of collaborative machine learning using 5G vehicular communications. IEEE Commun Surv Tutor. 2022;24(2):1280–303. doi:10.1109/COMST.2022.3149714.
9. Zhang Z, Xu C, Liu K, Xu S, Huang L. A resource optimization scheduling model and algorithm for heterogeneous computing clusters based on GNN and RL. J Supercomput. 2024;80(16):24138–72. doi:10.1007/s11227-024-06383-4.
10. Lee J, Cheng Y, Niyato D, Guan YL, González D. Intelligent resource allocation in joint radar-communication with graph neural networks. IEEE Trans Veh Technol. 2022;71(10):11120–35. doi:10.1109/TVT.2022.3187377.
11. He C, Lu Y, Ai B, Dobre OA, Ding Z, Niyato D. ICGNN: graph neural network enabled scalable beamforming for MISO interference channels. IEEE Trans Mob Comput. 2025:1–13. doi:10.1109/TMC.2025.3570648.
12. Abliz P, Ying S. Underestimation estimators to Q-learning. Inf Sci. 2022;607(3–4):173–85. doi:10.1016/j.ins.2022.05.090.
13. Shahgholi T, Khamforoosh K, Sheikhahmadi A, Azizi S. Enhancing channel selection in 5G with decentralized federated multi-agent deep reinforcement learning. Comput Knowl Eng. 2024;7(2):1–16. doi:10.22067/cke.2024.88900.1119.
14. Zhang X, Peng M, Yan S, Sun Y. Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. IEEE Internet Things J. 2020;7(7):6380–91. doi:10.1109/JIOT.2019.2962715.
15. Abbas F, Fan P, Khan Z. A novel low-latency V2V resource allocation scheme based on cellular V2X communications. IEEE Trans Intell Transp Syst. 2018;20(6):2185–97. doi:10.1109/TITS.2018.2865173.
16. Zhou Q, Yang S. Research on graph feature aggregation algorithm based on GCN and GAT. In: 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS); 2024 Jul 26–28; Shenyang, China. p. 964–70. doi:10.1109/ICPICS62053.2024.10795896.
17. Li H, Dou R, Keil A, Principe JC. A self-learning cognitive architecture exploiting causality from rewards. Neural Netw. 2022;150(1):274–92. doi:10.1016/j.neunet.2022.02.029.
18. Dubey AK, Jain V. Comparative study of convolution neural network's ReLU and leaky-ReLU activation functions. In: Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018; Singapore: Springer; 2019. p. 873–80. doi:10.1007/978-981-13-6772-4_76.
19. Liu S, Pan C, Zhang C, Yang F, Song J. Dynamic spectrum sharing based on deep reinforcement learning in mobile communication systems. Sensors. 2023;23(5):2622. doi:10.3390/s23052622.
20. Alsheikh MA, Hoang DT, Niyato D, Tan HP, Lin S. Markov decision processes with applications in wireless sensor networks: a survey. IEEE Commun Surv Tutor. 2015;17(3):1239–67. doi:10.1109/COMST.2015.2420686.
21. Ji M, Wu Q, Fan P, Cheng N, Chen W, Wang J, et al. Graph neural networks and deep reinforcement learning based resource allocation for V2X communications. IEEE Internet Things J. 2024;12(4):3613–28. doi:10.1109/JIOT.2024.3469547.
22. Ren J, Chai Z, Chen Z. Joint spectrum allocation and power control in vehicular communications based on dueling double DQN. Veh Commun. 2022;38(3):100543. doi:10.1016/j.vehcom.2022.100543.
23. Liu Z, Deng Y. Resource allocation strategy for vehicular communication networks based on multi-agent deep reinforcement learning. Veh Commun. 2025;53(2):100895. doi:10.1016/j.vehcom.2025.100895.
24. Gu X, Zhang G, Ji Y, Duan W, Wen M, Ding Z, et al. Intelligent surface aided D2D-V2X system for low-latency and high-reliability communications. IEEE Trans Veh Technol. 2022;71(11):11624–36. doi:10.1109/TVT.2022.3189627.
25. Liu P, Cui H, Zhang N. DDQN-based centralized spectrum allocation and distributed power control for V2X communications. IEEE Trans Veh Technol. 2025;74(3):4408–18. doi:10.1109/TVT.2024.3493137.

26. Zhou J, Liu L, Wei W, Fan J. Network representation learning: from preprocessing, feature extraction to node embedding. ACM Comput Surv. 2022;55(2):1–35. doi:10.1145/3491206.
27. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9). [Internet]. [cited 2025 Jun 11]. Available from: https://api.semanticscholar.org/CorpusID:16652630.
28. Ye H, Li GY, Juang B-HF. Deep reinforcement learning based resource allocation for V2V communications. IEEE Trans Veh Technol. 2019;68(4):3163–73. doi:10.1109/TVT.2019.2897134.
29. Chen Z, Alonso-Mora J, Bai X, Harabor DD, Stuckey PJ. Integrated task assignment and path planning for capacitated multi-agent pickup and delivery. IEEE Robot Autom Lett. 2021;6(3):5816–23. doi:10.1109/LRA.2021.3074883.
30. Qu N, Wang C, Li Z, Liu F, Ji Y. A distributed multi-agent deep reinforcement learning-aided transmission design for dynamic vehicular communication networks. IEEE Trans Veh Technol. 2023;73(3):3850–62. doi:10.1109/TVT.2023.3326877.