electronics

Article

An Enhanced MADDPG–A2C Framework for Optimized Resource Allocation in High-Speed Vehicular Networks

Linna Hu [1,*], Weixian Zha [1], Penghao Xue [1], Shuhao Xie [2], Bin Guo [3] and Wei Wang [1]

[1] School of Network and Communication Engineering, Jinling Institute of Technology, Nanjing 211169, China; 2507010001@stu.jit.edu.cn (W.Z.); 2407010018@stu.jit.edu.cn (P.X.); ww4546@jit.edu.cn (W.W.)
[2] School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1224014513@njupt.edu.cn
[3] College of Computer and Information Engineering, Xinjiang Agricultural University, Urumqi 830052, China; gb@xjau.edu.cn
* Correspondence: hulnnj@jit.edu.cn

Abstract

To address the degradation in communication performance caused by the high mobility and dynamic uncertainty in vehicular network channels, this paper proposes a hybrid resource allocation framework that integrates the advantage actor–critic (A2C) algorithm with the multi-agent deep deterministic policy gradient (MADDPG) algorithm. By modeling the high-speed vehicular network environment, the resource allocation task is formulated as a multi-agent deep reinforcement learning (MADRL) problem within a continuous action space. The proposed framework leverages the advantage function to refine gradient estimation, thereby improving training stability and convergence behavior. Additionally, regularization penalty terms and constraint mechanisms are incorporated into the learning process to balance multiple communication objectives. Specifically, the method aims to maximize the throughput of vehicle-to-infrastructure (V2I) links while ensuring the transmission reliability of vehicle-to-vehicle (V2V) links. In simulation experiments, the proposed method performs better in terms of convergence. Compared with the conventional MADDPG algorithm, the average access success probability is improved by 1.6%, and the average V2I throughput increases by 3.5%, indicating a significant enhancement in overall vehicular communication efficiency and transmission performance.

Keywords: vehicular network; resource allocation; A2C; MADDPG; MADRL

Academic Editor: Zuhaib Khan

Received: 13 February 2026
Revised: 6 March 2026
Accepted: 11 March 2026
Published: 13 March 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Advanced communication technologies are used by vehicular networks to propel advancements in intelligent mobility and autonomous driving [1]. The vehicular network achieves interconnectivity through vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-pedestrian (V2P) communication technologies, thereby providing a range of intelligent transportation services that cover various aspects including road safety, traffic management, and infotainment [2,3]. In a vehicular network, resource allocation (RA) is a key component in ensuring communication efficiency and network reliability. It involves various aspects, including communication resources, vehicular resources, computational resources, and storage resources. Vehicular network resource allocation is a complex process that involves multiple layers, various strategies, and specific

methods. It requires a comprehensive approach that considers the unique characteristics and requirements of the vehicular network environment, including channel dynamics, vehicle mobility, and the requirements for different quality of service (QoS) [4,5].

Traditional resource allocation methods often rely on centralized control strategies or predefined rules, which make it difficult to achieve ideal results in highly dynamic and uncertain vehicle networking environments. Conversely, reinforcement learning (RL) provides the mechanism for autonomous learning and decision-making, enabling vehicles to dynamically regulate their behavior based on changes in the environment and to find optimal resource allocation strategies. Traditional reinforcement learning, with algorithms like Q-learning and Sarsa at its core, primarily focuses on single-objective optimization for tasks such as spectrum assignment and power control in static or low-dynamic scenarios [6]. DRL addresses the dimensionality problem of traditional RL by introducing deep neural networks (DNNs), with typical algorithms including DQN (deep Q-network), DDPG (deep deterministic policy gradient), SAC (soft actor–critic), etc. MADRL is an extension and expansion of single-agent DRL in multi-agent interaction scenarios [7]. Through introducing interaction mechanisms and a distributed decision-making framework, MADRL demonstrates significant improvements compared with single-agent DRL in complex dynamic scenarios.

Despite the evolution of the resource allocation algorithm from static optimization to dynamic intelligent decision-making, which enhances complex environment adaptability, high mobility and heterogeneous requirements in vehicular networks remain critical challenges. Network resource scarcity and diverse service types further exacerbate the core dilemma in collaborative resource sharing between V2I and V2V links. Key challenges include responding to real-time network changes, increasing computational complexity (especially in multi-vehicle and multi-task scenarios); convergence instability in multi-agent algorithms within dynamic environments; and balancing fairness and efficiency via multi-objective optimization for diverse user/vehicle needs. Under this premise, reference [8] studied the issue of unfair network resource access caused by the uneven dwell time resulting from vehicle speed differences in the 5G NR V2I Mode 2 scenario. At the same time, it modeled and optimized the AoI, ensuring the timeliness of data transmission.

This paper focuses on communication resource allocation (RA) strategies for vehicular networks based on MADRL and proposes an improved joint optimization strategy for RA. The algorithm combines the MADDPG algorithm with the A2C framework to achieve efficient resource management in dynamic vehicular networks. The contributions are as follows:

(1) Construct a high-speed dynamic model of the vehicle network environment and formulate the RA problem as an MADRL problem in a continuous action space. Establish an objective optimization problem and a reward function under multiple demand constraints, with the primary aim of maximizing V2I throughput and ensuring V2V link transmission stability, thereby fulfilling diverse QoS requirements.
(2) Develop an optimized algorithm framework to address the convergence issues that traditional reinforcement learning algorithms encounter in high-speed dynamic environments. The algorithm combines the MADDPG algorithm with the A2C architecture. By integrating the A2C framework to optimize gradient estimation, the advantage function is embedded into the loss functions of both the actor and critic and participating in gradient calculation. This approach reduces the mean squared error and enhances the algorithm's convergence and stability.
(3) Design a multidimensional reward function that encompasses not only V2I and V2V link rates but also includes regularization penalty terms. Set specific constraints to ensure the reward function adapts effectively to environmental changes, enhancing learning efficiency to fulfil the requirements of real-world V2V communication environments.

The remainder of this paper is organized as follows: Section 2 reviews the existing research in this field. Section 3 introduces the system framework and formulates the optimization problem. Section 4 elaborates on the proposed resource allocation optimization scheme. Section 5 presents and discusses the simulation results. Finally, Section 6 concludes the work and highlights possible directions for future improvement.

2. Related Work

Vehicular network resource allocation algorithms have gradually transitioned from static, model-driven traditional optimization methods to dynamic, data-driven reinforcement learning (RL) and deep reinforcement learning DRL technologies, and ultimately toward intelligent multi-agent collaborative decision-making. Recent studies have improved performance and adaptability by integrating edge computing and UAV-assisted architectures, while also considering convergence and generalization capabilities in dynamic environments.

The dual-depth Q-network (DDQN) in Reference [9] is mainly applicable to single-agent environments and lacks the ability to handle multi-agent interactions. It performs poorly in continuous action spaces and collaborative or competitive tasks. In Reference [10], machine learning algorithms are combined with Kube edge computing for dynamic task offloading in RA. S. Gyawali et al. addressed the issue of collaborative misbehavior detection systems in vehicular networks failing to accurately provide feedback on attack information by proposing a dynamic reputation update approach based on DRL in Reference [11]. An improved deep reinforcement learning (IDRL) method is introduced in Reference [12] to reduce the control overhead of connected vehicle systems. Aforementioned studies considered only single-agent reinforcement learning algorithms. Although these algorithms outperform traditional methods, they struggle to achieve global optimality in complex environments, especially in non-convex optimization tasks.

Due to the inherent limitations of single-agent reinforcement learning algorithms in handling multi-agent interaction tasks, research has gradually shifted towards MADRL. MADRL, as a branch of RL, is better suited to address problems in naturally distributed and multi-participant systems such as VANETs. Among them, the MADDPG algorithm stands out as a classic approach. In reference [13], Ergün modeled resource allocation as a multi-agent deep reinforcement learning (RL) model and introduced a distributed multi-agent deep deterministic policy algorithm (MADDPG), which enhanced the capacity of the V2I link and the transmission rate of V2V links, and optimized the spectrum utilization. In reference [14], an MADDPG algorithm based on distributed execution is proposed to address the non-stationarity caused by rapid channel variations in high-speed dynamic networks. This algorithm achieves effective sharing of spectrum resources between vehicle-to-vehicle (V2V) links and vehicle-to-infrastructure (V2I) links. Although this algorithm has a relatively low computational cost, it still exhibits deficiencies in convergence performance and batch sample processing capabilities. Reference [15] proposed an improved scheme for the multi-agent coordinated actor–vritic method to address the inherent non-stationarity problem. Reference [16] introduces a conceptually simple and lightweight framework, proposing asynchronous variants of four standard reinforcement learning algorithms that have achieved success in various continuous motion control problems. Reference [17] proposed an improved MADDPG algorithm, which adopts a separation architecture design of learning and execution to solve the relevant optimization problems. Nevertheless, maintaining good convergence and model generalization in dynamic environments remains a significant challenge for this algorithm. References [18–22] have optimized the original multi-agent reinforcement learning algorithms to handle tasks with varying requirements. Despite the progress made by recent MADRL studies in addressing multi-agent interaction challenges, several critical limitations remain

unresolved, particularly in dynamic complex environments, multi-agent coordination, and algorithmic convergence.

The integration of DRL with emerging architectures such as edge computing and UAV-assisted systems offers new opportunities for real-time and adaptive resource allocation. Edge servers can reduce processing delays by handling computation offloading near vehicles. References [23,24] leverage edge computing for both static and dynamic resource management in V2X networks. UAVs serve as mobile base stations to extend coverage in areas with limited infrastructure. References [25,26] utilize UAVs to enable reliable data communication within V2X systems. Integrating DRL with these capabilities allows for flexible resource scheduling in MADRL frameworks [27].

Achieving fast convergence, strong generalization, and low computational overhead is essential for DRL algorithms deployed in highly dynamic vehicular environments. Ahmed Manzoor and his team have proposed a secondary RA scheme using a dual-timescale approach to efficiently allocate idle resources among stationary and mobile vehicles. By integrating this mechanism with the QMIX algorithm, they have enhanced computational resource allocation [28]. The PPO algorithm, introduced in 2017, achieved a breakthrough in training stability through clipping of the objective function. However, it has limitations in multi-agent scenarios, including high sample complexity, slow convergence speed, and insufficient scalability. Particularly in large-scale vehicular networks, maintaining the trust region of multi-agent interactions leads to exponential growth in computational overhead [29].

M. Babaeizadeh et al. have proposed an A2C algorithm framework that, through parallel processing, supports multiple agents operating in different environmental instances simultaneously [30]. A2C addresses the high variance issue associated with the basic actor–critic algorithm by introducing a baseline to the calculation of expected rewards. Compared with traditional reinforcement learning algorithms, the A2C algorithm can effectively reduce the data correlation and update instability by using multiple learners. The on-policy characteristic of the A2C algorithm enables an immediate link between sample collection and policy updates, meeting the millisecond-level response requirements of vehicles. Compared to PPO and SAC, the A2C algorithm exhibits superior computational efficiency due to its concise architecture and synchronous update mechanism, which reduces latency in real-time decision-making and is well suited to vehicle system resource constraints. The MAPPO mainly utilizes the advantage function to clip the objective function to restrict policy updates. Although this ensures stability, it may overly restrict policy exploration, leading to slow convergence in complex dynamic environments. In terms of environmental stability, A2C optimizes reward bias through the advantage function, mitigating training fluctuations caused by vehicle dynamic topology, thereby overcoming the significant sampling bias of PPO and the excessive exploration characteristics of SAC. A2C has been extensively validated in vehicle network optimization, facilitating comparison with existing research baselines and ensuring the reliability of conclusions. However, it may also lead to low sample efficiency. As the number of vehicles rises, it will significantly reduce the convergence speed of the training process [31].

Furthermore, data compression technology is crucial for enhancing the efficiency of vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication. For instance, when confronted with the challenges of limited bandwidth and high data transmission demands, Y. Wiseman [32] proposed a two-step reduction method for the transmitted images, significantly reducing the data transmission volume required for the vehicle networking system. Table 1 is a comparison of the literature.

Table 1. Literature comparison.

Algorithm	Reference	Limitations	Environment	Convergence	Scalability
DQN	[6], 2024; [9], 2023; [12], 2023	Hard to handle continuous actions; unstable in high-dynamic vehicular networks	Single-agent; discrete spectrum allocation	Slow; unstable in complex scenarios	Limited scalability
DDPG	[5], 2023; [18], 2023; [23], 2020	Sensitive to hyperparameters; overestimation bias	Continuous power control; single-agent	Better than DQN but unstable under dynamic channels	Limited scalability to multi-agent
PPO	[4], 2023; [24], 2023; [25], 2024	High sampling cost; limited fairness; weak generalization in large-scale IoV	Single-agent; continuous and discrete hybrid resource allocation	Stable convergence, better than DDPG/DQN	Poor scalability to multi-agent
MAPPO	[2], 2025; [29], 2025; [33], 2024	High computational cost for centralized critic; bottleneck in large networks	Multi-agent; CTDE framework	Stable; faster convergence than MADDPG	Better than PPO but still limited in very large-scale IoV
MADDPG	[13], 2025 [15], 2017; [19], 2022; [20], 2020	Training instability due to non-stationarity; high variance	Multi-agent; CTDE with deterministic policies	Converges faster than DDPG but is less stable than MAPPO	Limited scalability when the number of agents increases
MADQN	[9], 2023; [17], 2022; [34], 2025	Limited to discrete action spaces; poor adaptability to mixed continuous/discrete tasks	Multi-agent; value-based	Converges more slowly than policy gradient methods	Moderate scalability but struggles in continuous IoV tasks
A2C	[16], 2016; [30], 2024; [31], 2024	Sample inefficiency; sensitive to learning rates	Single-agent; synchronous actor–critic	Faster convergence than vanilla policy gradient; less stable than PPO	Limited scalability; mostly used in small IoV environments

3. System Model and Problem Formulation

3.1. System Model

This research primarily investigates the vehicular communication environment under high-speed dynamic networks, considering the scenario depicted in Figure 1, which includes a base station and multiple vehicular users. These vehicles are categorized as V2I users and V2V users depending on their communication requirements. The V2I link enables communication between base stations, vehicles, and roadside units, while the V2V links facilitate direct communication between the current vehicle and neighboring vehicles. The set of V2I vehicles links is denoted as $S_{V2I} = \{1, 2, \ldots, M\}$, and the set of V2V vehicles links is denoted as $S_{V2V} = \{1, 2, \ldots, N\}$. To conserve spectral resources, multiple V2V users can reuse the spectrum resources of V2I links to achieve sharing.

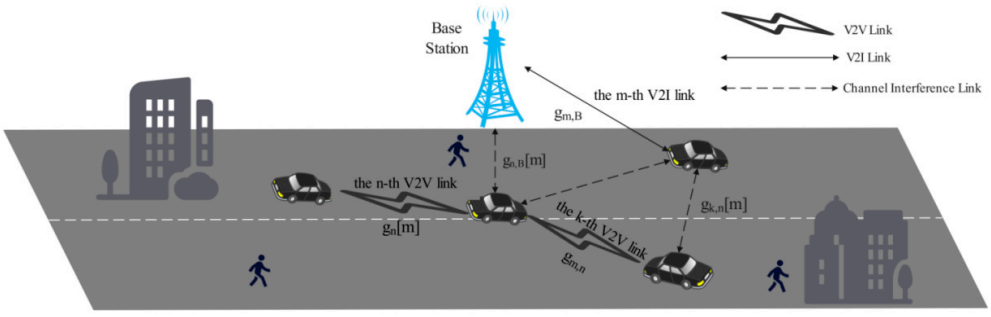


Figure 1. System model of the vehicular network.

3.2. Communication Model

In the vehicular network environment constructed in this paper, it is presumed that the fading characteristics of each orthogonal sub-channel are identical and independent of each other. The total spectral resources of this uplink channel are evenly divided among M orthogonal sub-channels. To allocate the V2I link spectral resources more reasonably and efficiently, thereby fully utilizing the channel conditions, this paper employs a fully distributed resource allocation mechanism. It is assumed that all vehicular user transceivers employ a single antenna, and that each V2I vehicular user has exactly one V2I uplink channel for communication. The communication model considers slow fading caused by building blockage and changes in path length, as well as fast fading caused by multipath propagation, and both phenomena coexist.

3.2.1. Interference Analysis

In the vehicular networking communication system, various interference sources will have a serious impact on the communication quality. On sub-channel m, the interference from the n-th V2V link is denoted by $I_n^v[m]$, and from the rest of the V2V links, it is represented by $I_k^v[m]$:

$$I_n^v[m] = \sum_{n=1}^{N} \delta_n[m] p_n^v[m] g_{n,B}[m] \tag{1}$$

$$I_k^v[m] = \sum_{k \neq n}^{N} \delta_k[m]\, p_k^v[m] g_{k,n}[m] \tag{2}$$

where $p_n^v[m]$ represents the transmission power of the n-th V2V link, $g_{n,B}[m]$ denotes the channel gain from the V2V link transmitter to the base station, $\delta_n[m]$ indicates whether the m-th V2I link is reused by the n-th V2V link, with a value of 1 if it is reused and 0 otherwise, and $g_{k,n}[m]$ represents the channel gain of the k-th V2V link.

The interference by the m-th V2I link is denoted by $I_m^I[m]$:

$$I_m^I[m] = P_m^I[m] g_{m,n}[m] \tag{3}$$

The SINR for the m-th V2I link and the n-th V2V link are, respectively, expressed as

$$\gamma_m^I[m] = \frac{p_m^I[m] g_{m,B}[m]}{I_n^v[m] + \sigma^2} \tag{4}$$

$$\gamma_n^v[m] = \frac{p_n^v[m] g_n[m]}{I_m^I[m] + I_k^v[m] + \sigma^2} \tag{5}$$

where σ^2 indicates the power of Gaussian white noise, and $g_n[m]$ signifies the channel gain of the n-th vehicle on the m-th orthogonal sub-channel.

3.2.2. Throughput Analysis

According to Shannon's formula [33], the total throughput of the V2I and V2V is derived, respectively, as $R_m^I[m]$ and $R_n^v[m]$:

$$R_m^I[m] = Blb(1 + \gamma_m^I[m]) \tag{6}$$

$$R_n^v[m] = Blb(1 + \gamma_n^v[m]) \tag{7}$$

where B represents the channel bandwidth.

Since the travel speed and traffic position status of V2V communication users are key to service safety information, ensuring the effective transmission probability of V2V is crucial. The effective transmission probability ψ_n is described as follows:

$$\psi_n = p_r\left\{\sum_{t=1}^{T}\sum_{m=1}^{M}\delta_n[m]R_n^v[m,t] \geq \frac{Bp}{\triangle T}\right\}, n \in N \tag{8}$$

Among them, it should be noted that Bp indicates the size of the generated V2V payload, t represents the time slot, and $\triangle T$ denotes the coherence time.

3.3. Problem Formulation

The purpose of this paper is to improve the overall throughput of V2I links while ensuring the transmission reliability of V2V links, focusing on the average transmission success probability of V2V links. By considering the key quality of service constraints such as delay and channel interference, the following reliability assurance optimization problems are formulated:

$$OP1: max_{(p_n^v[m],p_m^I[m],\delta_n[m])}\sum_{m=1}^{M}R_m^I[m] \tag{9}$$

$$OP2: \max_{(p_n^v[m],p_m^I[m],\delta_n[m])}\mathbb{E}_{\boldsymbol{h}}\left[1-\frac{n'}{N}\right] \tag{10}$$

where n' denotes the number of V2V links that fail to satisfy the minimum payload transmission requirement within the coherence time $\triangle T$, defined as

$$n' = \sum_{n=1}^{N}\mathbb{I}\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\delta_n[m]R_n^v[m,t] < \frac{B_p}{\Delta T}\right) \tag{11}$$

In Equation (11), $\mathbb{I}(\cdot)$ represents the indicator function that equals 1 when the condition is satisfied (i.e., the n-th V2V link fails to transmit the required payload Bp within $\triangle T$) and 0 otherwise. $\mathbb{E}_{\mathbf{h}}[\cdot]$ denotes the expectation over all possible channel states.

Remark 1: The objective function in OP2 is mathematically equivalent to maximizing the average access success probability of V2V links. Specifically,

$$\mathbb{E}_h\left[1-\frac{n'}{N}\right] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_h\left[\mathbb{I}\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\delta_n[m]R_n^v[m,t] \geq \frac{B_p}{\Delta T}\right)\right] = \frac{1}{N}\sum_{n=1}^{N}\psi_n \tag{12}$$

where ψ_n is the effective transmission probability of the n-th V2V link as defined in Equation (8). Therefore, OP2 directly corresponds to constraint C4 ($\psi_n \geq \rho_0$), ensuring the transmission reliability of V2V links.

$s.t.$

$$C1: 0 \leq p_m^I[m] \leq p_{max}; 0 \leq p_n^v[m] \leq p_{max}; \forall m \in M, \forall n \in N \tag{13}$$

$$C2: 0 \leq I_n^v[m] \leq I_{max}; 0 \leq I_m^I[m] \leq I_{max}; 0 \leq I_k^v[m] \leq I_{max};\ \forall m \in M, \forall n \in N \tag{14}$$

$$C3: R_m^I[m] \geq R_{min};\ \forall m \in M \tag{15}$$

$$C4: \psi_n \geq \rho_0;\ \forall m \in M, \forall n \in N \tag{16}$$

$$C5: \tau_n^v \leq \triangle T;\ \forall n \in N \tag{17}$$

$$C6: \delta_n[m] \in \{0,1\};\ \forall m \in M, \forall n \in N \tag{18}$$

$$C7: R_n^v[m] \le R_{max}; \forall n \in N \tag{19}$$

where N represents the total number of V2V communication links. C1 represents the maximum permitted transmission power. C2 represents the limitations of interference at the receiving end of the links. C3 represents the minimum throughput constraint that the V2I link can tolerate. C4 represents the effective transmission probability constraint, which is consistent with the objective of OP2. C5 represents the maximum transmission delay allowed. C6 represents whether the spectrum resources of the V2I link have been reused, which is a binary variable. C7 represents the maximum throughput constraint that the V2V link can tolerate, that is, the maximum transmission rate constraint.

4. Proposed Resource Allocation Optimization Strategy Based on MADDPG and A2C

Building upon an in-depth study of the MADDPG and A2C algorithms, an improved MADDPG-A2C joint optimization algorithm for resource allocation has been proposed in this paper. The proposed strategy adopts the centralized training with decentralized execution (CTDE) framework of MADDPG to address multi-agent collaborative decision-making problems. It introduces the advantage function mechanism of A2C to optimize the gradient estimation process. By incorporating the advantage function from A2C into the actor and critic loss functions of MADDPG through multiplicative weighting, it enables direct participation in gradient calculations, thereby addressing issues of unstable training and large gradient variance in traditional MADDPG. The combination of the MADDPG and A2C algorithms provides an effective learning framework for multi-agent systems, enabling better strategy learning and resource allocation under dynamic and complex environmental conditions.

4.1. State Space Design

In time slot t, the environmental state that agent n currently encounters is denoted as s_t^n. Agent n then produces the corresponding action A_t^n according to the current policy π. For the vehicular network as a whole, the joint state space formed by all agents is denoted by S_t, and the joint action A_t is generated under the same policy π, resulting in a joint reward R_t after executing the actions together. In the neural network, when the agent n of the state s_t^n transits to the next state, it is denoted as s_{t+1}^n, with probability ρ_t^n. For each independent agent, taking agent n as an example, it can only obtain partial state information and the action space of its own environment, but cannot obtain global state information and the action space information of other agents, which are all unknown factors. The expression for ρ_t^n is as follows:

$$\rho_t^n = p_r\left(\frac{s_t^n}{s_{t+1}^n}, A_t^n\right) \tag{20}$$

The expression for s_t^n is

$$s_t^n = \left(g_n[m], g_{k,n}[m], g_{n,B}[m], g_{m,n}[m]\right) \tag{21}$$

The state space inherently includes dynamic information such as vehicle position and speed through real-time updated channel gain parameters. At the beginning of each time slot, the system first calculates and updates the path loss and small-scale fading parameters of all links according to the real-time position coordinates and moving speed of vehicles, and then uses the updated channel gains as the state input, ensuring that the state

representation can fully reflect the time-varying characteristics of high-speed dynamic networks.

4.2. Action Space Design

The vehicular network resource allocation problem primarily concerns the optimization of orthogonal sub-channel allocation for V2V communication and transmission power. The spectrum resources of each V2I link are evenly divided into M orthogonal sub-channels. The n-th V2V link can select one channel from these M orthogonal sub-channels for sharing and adjust their transmission power. The boundaries of the continuous power value $p_n^v[m]$ are defined as $0 \le p_n^v[m] \le P_{max}$. The action selected by agent n is given by

$$A_t^n = \{p_n^v[m]\} \tag{22}$$

4.3. Reward Function Design

The optimization problem is established in a multi-objective optimization framework and is converted into a single-objective problem through a weighted linear combination. In a unit time slot t, the objective function treats completed transmissions and unfinished transmissions as two separate cases. The first objective function is the reward for the V2I link, defined as the total throughput of the V2I link. For V2V users, at time t, the reward related to unfinished V2V transmissions is defined according to the payload that has been delivered. The second objective function encourages V2V users to increase the transmission rate by assigning a scalar value that is larger than the effective V2V transmission rate. The two objective functions can be written as follows:

$$Re_{V2I} = \sum_{m=1}^{M} R_m^I[m] \tag{23}$$

$$Re_{V2V} = \begin{cases} \sum_{m=1}^{M} \delta_n[m] R_n^v[m,t] \,, B \ge 0 \\ \Phi \,, B < 0 \end{cases} \tag{24}$$

Then, by applying weight coefficients, the reward function is obtained as follows:

$$Re_{total} = \lambda * Re_{V2I} + (1-\lambda) * \sum_{n=1}^{N} Re_{V2V} - \varphi \tag{25}$$

In Equation (25), λ represents the weight coefficient, and it is within the range (0, 1), indicating the importance level of this element's reward. φ is the penalty factor, which is 0 when certain constraint conditions are met and ζ when the constraints are not met. The expression for ζ is as follows:

$$\zeta = \chi * \frac{Re_{V2V}}{\eta} \tag{26}$$

In Equation (26), χ is an adjustable parameter used to control the degree of the penalty. η represents the logarithm of the number of communication links.

To maintain the stability of rewards while accounting for current rewards and expected future rewards, this algorithm aims to determine the optimal strategy $\pi*$ for maximizing the cumulative expected reward; that is,

$$Re_{max} = E_\pi \left[\sum_{\tau=0}^{\infty} \varpi^\tau] Re_{t+\tau} \right]; 0 \le \varpi \le 1 \tag{27}$$

https://doi.org/10.3390/electronics15061214

4.4. Joint Optimization Algorithm Based on MADDPG and A2C

Aiming to combat the inevitable multi-agent concurrent behaviors in the traditional MADDPG algorithm, which lead to the drawbacks of low sample efficiency and unstable training, an improved cooperative optimization architecture integrating MADDPG and A2C is proposed, as shown in Figure 2. Figure 2 illustrates the system architecture and the reinforcement learning operation process. Specifically, the left side of the figure presents the V2I/V2V hierarchical communication scenario, while the right side shows the operational flow of the MADDPG-A2C multi-agent learning framework. Therefore, this diagram comprehensively presents the core design of this study from two dimensions: the communication layer and the reinforcement learning framework operation process.

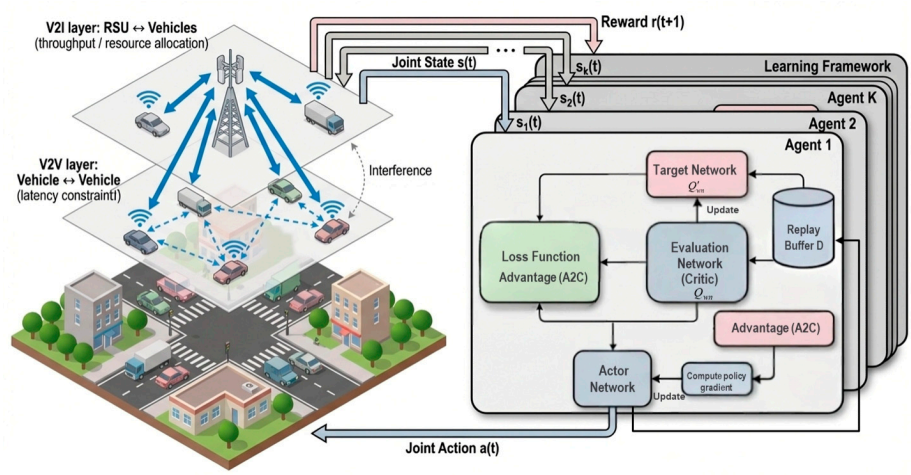


Figure 2. System architecture and operation flowchart.

To this end, traditional MADDPG integrates the A2C framework, leveraging the classical advantage function to compute policy gradients while updating the loss functions of both actor and critic networks. Accordingly, this approach minimizes mean squared error, simplifies handling of state spaces and action spaces, and effectively boosts system sampling efficiency and performance stability.

For clarity, the overall framework of the optimized algorithm and the system process are shown in Figure 3. The algorithm in this study adopts centralized training and distributed execution. Each independent agent n, where $n \in (1, 2, \ldots, N)$, outputs the corresponding action A_t^n through the actor network according to its own state s_t^n. There are N independent actor networks for N agents, corresponding to N policy functions and N critic networks. The policy function of the actor network is denoted as π_n policy. For a given critic, the input contains the outputs of N actor networks and finally provides feedback, with the feedback value denoted as Q_n. As training proceeds over multiple iterations, the global optimal solution is eventually derived, maximizing the reward return and causing the reward value to gradually converge. In summary, the improved MADDPG-A2C algorithm introduces the classic advantage function of the A2C algorithm into the original MADDPG framework to reduce mean square error and enhance efficiency and stability.

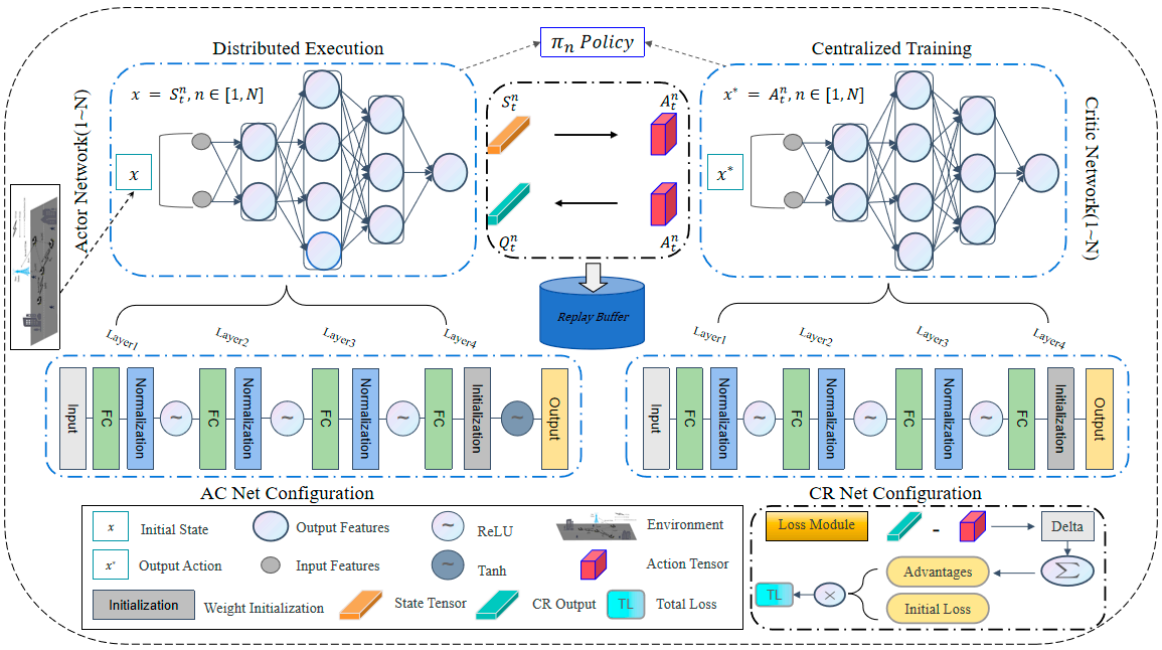


Figure 3. MADDPG-A2C algorithm framework.

In the MADDPG-A2C framework, two typical network architectures are used, namely, the actor and critic. The actor network is used to collect the corresponding environmental states and then execute the corresponding actions according to policy π, while the critic network uses the state–action value function to assess the quality of action A_t^n according to the policy. The current environmental state of agent n is s_t^n, and γ is the discount factor of rewards, which ranges from 0 to 1. In implementation, the continuous action variable space is discretized into four normalized levels, i.e., [−10, 5, 15, 23] dBm [34,35]. Therefore, the state–action value function of MADDPG-A2C is as follows:

$$Q_n(s_t^n, A_t^n) = E[R_t^n + \gamma Q(s_{t+1}^n, A_{t+1}^n)] \tag{28}$$

The optimal action plan is derived under the optimal strategy π_n to maximize the expected return. The expressions are as follows:

$$Actor_{Loss} = -E[Q_n(s_t^n, A_t^n|\omega) * advantages] \tag{29}$$

$$Critic_{Loss} = E\left\{\left[\left(y - Q_n(s_t^n, A_t^1, \ldots\ldots, A_t^N)\right) * advantages\right]^2\right\} \tag{30}$$

where y represents the target Q-value, and the advantage function is given by

$$A(s_t^n, A_t^n) = Q(s_t^n, A_t^n) - V(s_t^n) \tag{31}$$

The essence of the advantage function $A(s_t^n, A_t^n) = Q(s_t^n, A_t^n) - V(s_t^n)$ is to eliminate the state value function $V(s_t^n)$ as a baseline, thereby reducing the variance in the policy gradient estimation. In the standard policy gradient theorem, the policy gradient can be expressed as

$$\nabla J(\theta) = E[\nabla log\pi\theta(s_t^n|A_t^n) * Q(s_t^n, A_t^n)] \tag{32}$$

After introducing the advantage function, the policy gradient becomes

$$\nabla J(\theta) = E[\nabla log\pi\theta(s_t^n|A_t^n) * A(s_t^n, A_t^n)] \tag{33}$$

Here, $A(s_t^n, A_t^n) = Q(s_t^n, A_t^n) - V(s_t^n)$. Since E $[A(s_t^n, A_t^n))|s_t^n] = 0$, the advantage function maintains the unbiasedness of the gradient and significantly reduces the variance, as $\mathrm{Var}[A(s_t^n, A_t^n)] \leq \mathrm{Var}[Q(s_t^n, A_t^n)]$.

A lower-variance gradient leads to more stable learning dynamics with fewer oscillations, while the centered signal $A(s_t^n, A_t^n) = Q(s_t^n, A_t^n) - V(s_t^n)$ emphasizes the relative action quality rather than state difficulty. As a result, the actor can make more consistent progress and achieve faster practical convergence under the same training budget and is less likely to take noisy steps that cause oscillations or temporary performance drops; consequently, the learning curve becomes more stable and empirically closer to a monotonic improvement trend.

When the $Critic_{Loss}$ is continuously differentiable, the parameters ω can be updated through gradient descent. Here, the update strategy of the A2C algorithm is adopted as

$$\omega_n \leftarrow \omega_n + \beta_{critic} \nabla V_{\omega_n}(s_t^n) * \varepsilon \tag{34}$$

where β_{critic} represents the learning rate and ε is the Temporal Difference (TD) error. Each agent modifies the evaluation network parameters of the actor by maximizing the objective function when it seeks to maximize the cumulative reward. The objective function is given by

$$J_\theta = E[Q_n(s_t^n, A_t^n)] \tag{35}$$

Similarly, J_θ is continuously differentiable, allowing for gradient updates to the parameters θ. The update strategy of the A2C algorithm is adopted as

$$\theta_n \leftarrow \theta_n + \alpha_{actor} \nabla_{\theta_n} log \pi_{\theta_n}(s_t^n | A_t^n) * \varepsilon \tag{36}$$

where α_{actor} is the learning rate of the actor network. ε is TD error. The expression for the soft update is as follows:

$$\theta_n' \leftarrow \tau * \theta_n + (1 - \tau) * \theta_n' \tag{37}$$

$$\omega_n' \leftarrow \tau * \omega_n + (1 - \tau) * \omega_n' \tag{38}$$

The agent generates actions through an independent actor network. The parameters of the critic network and actor network adopt the synchronous update strategy of A2C, combined with the Temporal Difference (TD) error to achieve efficient parameter iteration. The target network still follows the soft update mechanism of MADDPG to ensure training stability. After being centrally evaluated and fed back by their respective critic networks, the networks are updated and iteratively trained in combination with the A2C advantage function to converge to the global optimal state. Algorithm 1 shows the pseudocode for the proposed MADDPG-A2C algorithm, and Figure 4 shows the flowchart of the algorithm.

In the MADDPG framework, the critic network needs to handle the global state and all agents' actions, and its input dimension grows linearly with the number of agents N. The advantage function calculation introduced by MADDPG-A2C involves additional state value estimation, but this operation is of the same order as the forward propagation of the critic network. Therefore, the asymptotic complexity of MADDPG-A2C is at the same level as that of MADDPG. MADDPG-A2C adds the estimation of the state value function V(s) on top of MADDPG. The structure of the V network is of the same order as the critic, with a computational complexity of O(Nd·d) = O(Nd²), and the calculation of the advantage function A = Q − V is an element-wise operation of O (1). Therefore, the total complexity remains O(N·d²), where N is the number of agents and d is the dimension of the state space (assuming that the state dimension d_s, action dimension d_a, and hidden-layer dimension h are of the same order).

The proposed MADDPG-A2C algorithm is described in Algorithm 1.

Algorithm 1. Proposed MADDPG-A2C Algorithm.

1. Initialize Actor Network θ_n and Critic Network ω_n
2. Initialize Target Actor Network θ_n' and Target Critic Network ω_n'
3. Initialize ReplayBuffer
4. Setting hyperparameters such as learning rate and target Network update rate
5. ***For*** episode = 1 to M do
6. Initialize environment state
7. Updating vehicle position and large-scale fading#
8. ***For*** t = 1 to max-episode-length do
9. For each agent n, the action $A_t^n = \mu_{\theta_n}(Q_n) + \epsilon$ is selected, where ϵ is the exploration noise.
10. Executing the action A_t^n and observing the received reward r_n and the next state s_{t+1}^n
11. Storing the tuple $(s_t^n, A_t^n, r_n, s_{t+1}^n)$ into the experience replaybuffer D
12. $s_t^n \leftarrow s_{t+1}^n$
13. ***For*** agent n = 1 to N do
14. Sampling a batch of data at random from the experience replaybuffer D.
15. Defining the target Q-value as $y = r_n + \gamma Q_{t+1}(s_{t+1}^n, A_{t+1}^1, \ldots\ldots, A_{t+1}^N)$, where γ is the discount factor.
16. Using the Critic Network to estimate the current state value $V(s_t^n)$ and the state-action value $Q_n(s_t^n, A_t^n)$, and calculate the Temporal Difference error ε. $\varepsilon = r_n + \quad \gamma V_{\omega_n}(s_{t+1}^n) - V_{\omega_n}(s_t^n)$.
17. Update the Critic Network parameters ω_n to minimize the TD error.
18. Define the advantage function Equation (31).
19. Minimize the loss according to Equation (30).
20. Compute the policy gradient of the Actor Network: $\nabla_{\theta_n} J(\theta_n) = E\left[\nabla_{\theta_n} log\ Q_n(s_t^n, A_t^n) A(s_t^n, a_t^n)\right]$.
21. Update the Actor Network parameters θ_n according to Equation (36).
22. Update the Critic Network parameters ω_n according to Equation (34).
23. Update the Target Network parameters using the soft update method:# Equation (37), Equation (38).

End if

End for

End for

https://doi.org/10.3390/electronics15061214

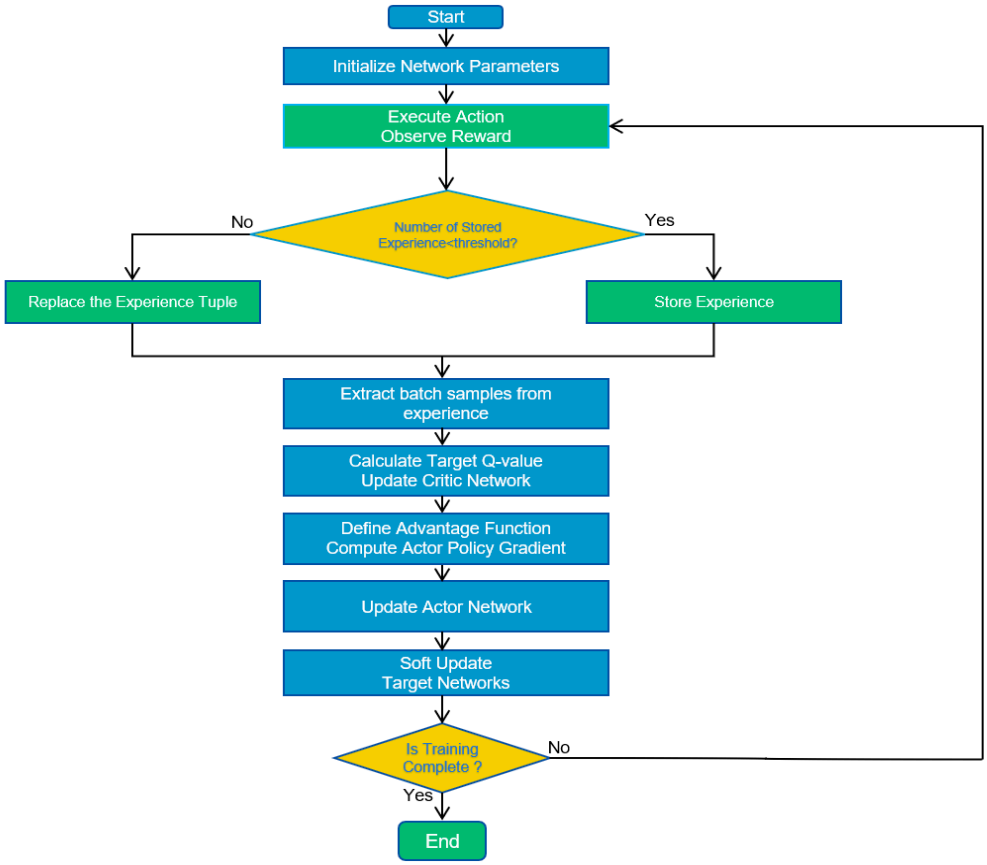


Figure 4. The proposed MADDPG-A2C algorithm flowchart.

The proposed framework is built upon the MADDPG architecture. The advantage function is used to refine the gradient estimation and reduce the mean squared error. It is directly embedded into the loss functions of the actor and critic through multiplicative weighting. This design enables the advantage function to directly participate in the gradient calculation, thereby improving the estimation accuracy. At the same time, the advantage function in the framework can directly correct the Q-value gradient to quickly adapt to the changes in signal-to-noise ratio caused by high-speed movement. Moreover, the proposed framework uses the advantage function to weight multidimensional rewards, ensuring the reliability of V2V transmission while maximizing V2I throughput, avoiding the inherent over-conservative problem of the MAPPO clipping mechanism.

In the proposed centralized training and decentralized execution framework, the global observations required by the centralized critic are aggregated at the edge side by an RSU or a BS. The aggregated information includes channel and interference measurements, resource occupancy, and link-level QoS states such as remaining payload and remaining deadline. These observations are used to train the centralized critic together with the joint actions stored in the replay buffer. During online execution, each vehicle runs only the decentralized actor and makes decisions based on local observations. Therefore, per-slot, low-latency decisions do not rely on real-time global information exchange, while the critic-related aggregation is confined to the training phase and can be performed at a slower timescale.

5. Experimental Analysis and Discussion

5.1. Simulation Environment Setup

An urban vehicular network simulator based on 3GPP TR 37.885 [35] was implemented in Python 3.7 (TensorFlow 1.14.0) and run on a single machine equipped with an NVIDIA RTX 4060 (8 GB VRAM). Bandwidth was set to 4 MHz based on 3GPP V2X side link recommendations (1.4–20 MHz) and literature [35,36], balancing multi-agent communication needs and computational complexity. The Adam optimizer with learning rate 0.001 was selected via grid search over $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ to balance convergence speed and stability in multi-agent settings, leveraging deep reinforcement learning actor–critic tuning experience. Assuming a one-to-one correspondence between V2V and V2I links, different sizes of V2V link loads are used to confirm the robustness of the algorithm. Table 2 presents the simulation parameters.

Table 2. Simulation parameters.

Vehicular Environment Parameters	Value
Carrier frequency f_c	2 GHz
Number of V2I links M	4
Number of V2V links N	4
Bandwidth B	4 MHz
Noise power N_0	−114 dBm
Base station antenna height	25 m
Base station single-antenna gain g	8 dBi
Base station noise figure	5 dB
Vehicle noise figure	9 dB
Vehicle speed V	5–50m/s
Time constraint	100 ms
Max transmission power of the V2I P_{max}^{V2I}	0.199 w
Max transmission power of the V2V P_{max}^{V2V}	0.199 w
Transmission power of the V2I P^{V2I}	[−10, 5, 15, 23] dBm
Vehicle antenna height	1.5 m
V2V link path loss model	$22.7\lg(d) + 41 + 20\lg(f_c/5)$
V2I link path loss model	$128.1 + 37.6\lg(d)$
Fast-fading update	1 ms

The maximum V2V transmission power is divided into specific levels to better control interference and facilitate quantization during the algorithm implementation and simulation, and to reduce the computational complexity. V2I link models mainly focus on the impact of distance. V2V communication is usually carried out in a higher frequency range (such as on-board Wi-Fi, LTE, etc.), and the model should consider not only distance but also the impact of frequency on path loss. Therefore, the $128.1 + 37.6\lg(d)$ model is selected for V2I links, and the $22.7\lg(d) + 41 + 20\lg(f_c/5)$ model is selected for V2V links. Based on actual application scenarios, the vehicle speed V was set to 5–50 m/s, and the V2I transmission power and path loss were modeled as dynamic variables.

5.2. Ablation Experiment

We verify the contributions of three reinforcement learning algorithms (DDPG, A2C, and MADDPG) and the improved reward function (Reward*) by comparing the performance metrics of various component combinations. Table 3 presents the performance metrics (reward, V2V success probability, V2I throughput, and MFLOPS) of five experimental groups with different method combinations.

Table 3. Comparison of ablation experiment results.

Index	Methods				Reward	V2V Links' Success Probability	V2I Links' Throughput (Mbps)	MFLOPS (M)
	DDPG	A2C	MADDPG	Reward*				
1	√				64.47	0.688	27.43	**0.103**
2		√			85.97	0.8217	28.61	0.108
3			√		87.59	0.837	29.8	0.43
4		√	√		87.72	0.84	29.75	0.516
5		√	√	√	**88.75**	**0.851**	**30.84**	0.516

Reward* denotes the improved reward function, √ indicates that this module has been used and bold black indicates the best performance metric.

The ablation experiment demonstrates that the MADDPG-A2C strategy proposed in this paper, which is a combination of A2C, MADDPG, and an improved reward function, is the optimal configuration. It achieves the highest reward value (88.75), vehicle-to-vehicle communication success probability (0.851), and vehicle-to-infrastructure communication throughput (30.84 Mbps). This confirms that multi-agent collaboration (MADDPG) and single-agent optimization (A2C) have a synergistic effect and that the improved reward function effectively enhances system performance without increasing additional computational costs. Single-algorithm groups (DDPG and A2C) have lower MFLOPS but poorer performance, while multi-algorithm combinations improve performance at the cost of higher computational load.

5.3. Comparative Experimental Analysis

The comparative experiment mainly compares different algorithms in terms of reward, average throughput, average access success probability, and total throughput. The reward comparison under different algorithms is depicted in Figure 5. Semi-transparent shading is used to represent the 95% confidence interval (CI) ranges. The figure illustrates the gradual convergence of reward values for different algorithms with the increase in training rounds. The MADDPG-A2C algorithm proposed in this paper converges the fastest. Compared with other algorithms, MADDPG-A2C has the narrowest 95% confidence interval, indicating a smaller sampling error in its mean estimation and higher reliability. This is because the MADDPG-A2C algorithm proposed in this paper introduces an appropriate advantage function, which makes the reward feedback starting point higher, and at the same time, it can become stable and converge faster under fewer training episodes.

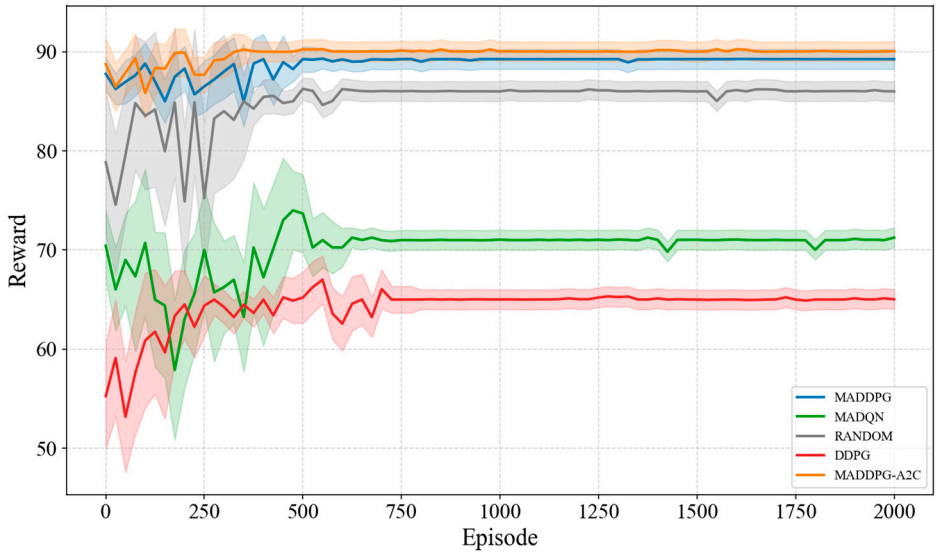


Figure 5. Comparison of reward under different algorithms.

Figure 6 illustrates the impact of the number of training rounds under different algorithms on the average V2I link throughput. The shaded area in the figure represents the 95% confidence interval. When the training set size is small, the average V2I throughput obtained by each algorithm fluctuates significantly, and the confidence interval (CI) width is relatively wide. As the size of the training set increases gradually, the average throughput shows an upward trend and eventually converges, with the 95% confidence interval narrowing. Among them, the MADQN algorithm converges the fastest. Simultaneously, it is evident that under the same number of training sets, the proposed MADDPG-A2C algorithm can obtain a higher average V2I link throughput. This is mainly due to the fact that the neural network in the MADQN algorithm contains only one Q-network, and the DDPG algorithm adopts single-agent training and learning, which cannot obtain the global optimal solution, so the rate is low. The RANDOM algorithm can avoid corresponding errors when exploring unknown environments and help to avoid the model falling into local optimal solutions. Theoretically, its performance is better than that of MADQN and DDPG. However, the RANDOM algorithm is generally suitable for dealing with simple environments.

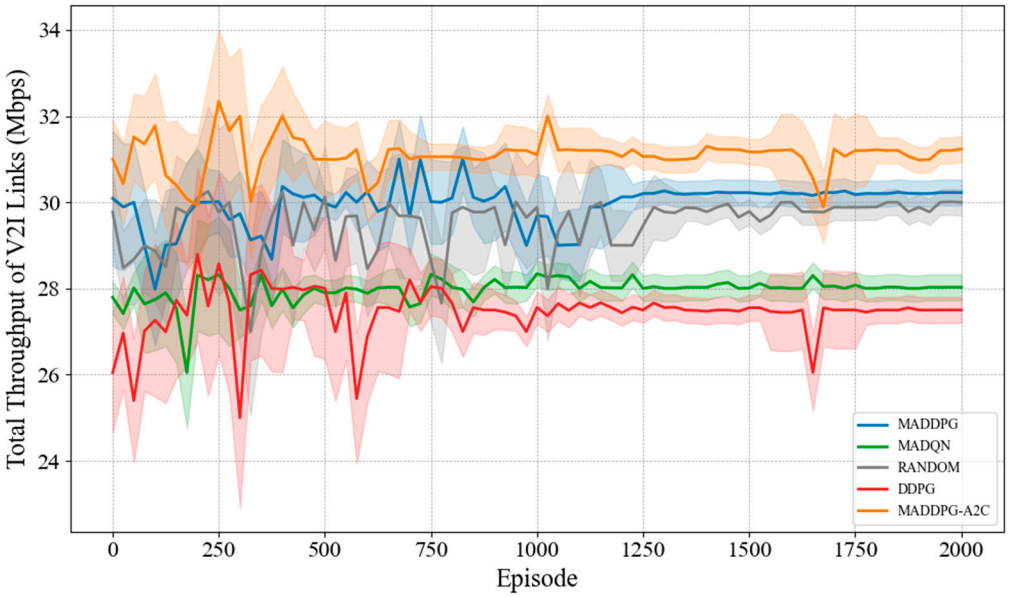


Figure 6. Comparison of the average throughput of V2I links under different algorithms.

Figure 7 shows the change in the average access success probability of V2V links under different training episodes. Among the five algorithms, the proposed MADDPG-A2C algorithm maintains a relatively high success probability at around 0.85 under different training episodes, which is relatively stable, and its access success probability does not fluctuate significantly with the change in the number of episodes. Among the five algorithms, the DDPG algorithm uses a single agent, so the success probability is relatively low, and the fluctuation is obvious. Figure 7 illustrates that the proposed algorithm has excellent performance.

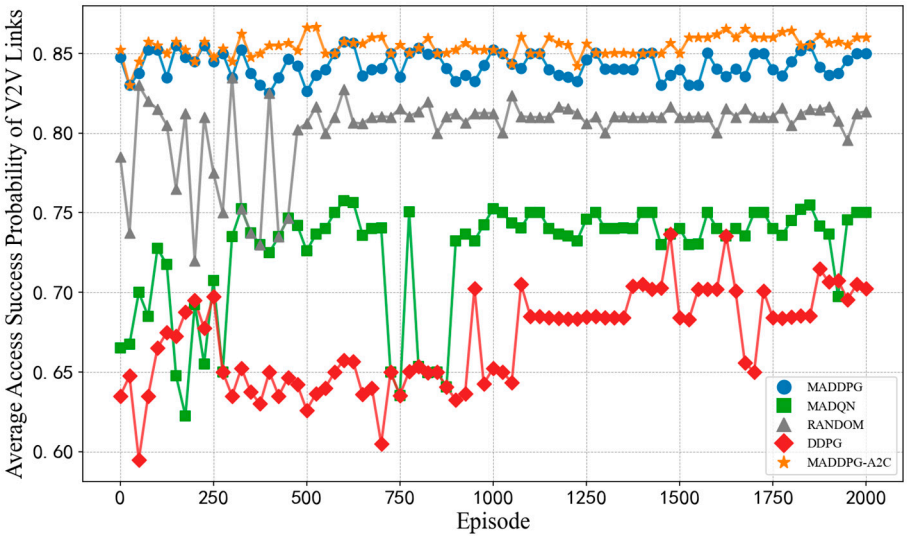


Figure 7. Comparison of the average access success probability of V2V links.

Figure 8 reflects the relationship between V2V load size and the average success probability of V2V transmission. The success probabilities for all five algorithms decrease as the V2V load size increases, with the proposed algorithm showing the slowest decline. To exclude minimal probability effects, a load peak of 6 × 1060 Bytes is considered. At a load size of 1 × 1060 Bytes, all five schemes exhibit high success probabilities, with the proposed algorithm achieving the highest. At a load size of 3 × 1060 Bytes, the proposed algorithm outperforms MADDPG, MADQN, RANDOM, and DDPG by 18.1%, 85.7%, 54.7%, and 47.7%, respectively. Experimental results demonstrate that the MADDPG-A2C algorithm consistently maintains the highest average success probability across varying V2V load sizes.

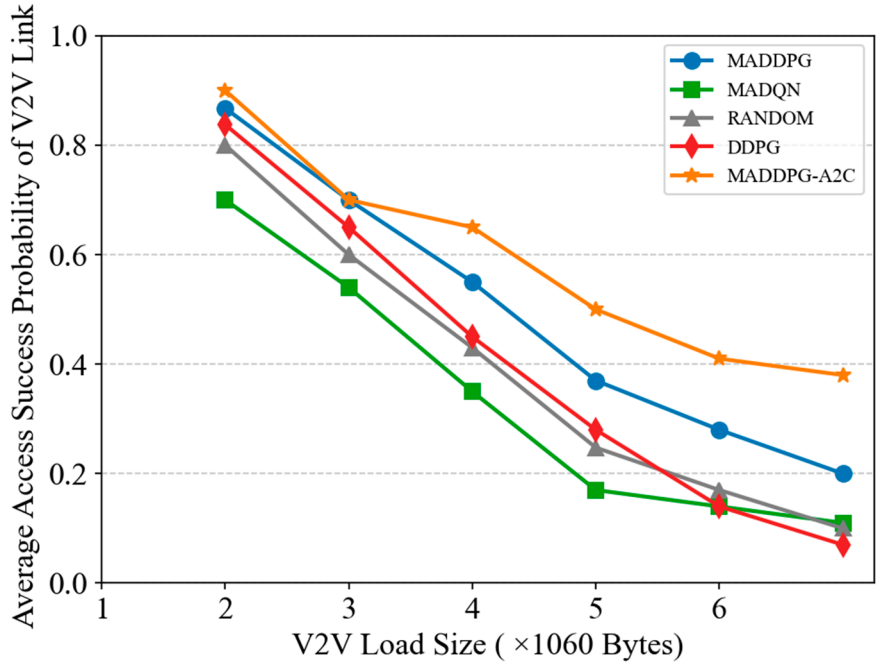


Figure 8. Comparison of the average access success probability of V2V links with load size.

Figure 9 shows the relationship between the variation in V2V load size under various methods and the total V2I throughput. In Figure 9, vertical error bars are added at each

load size point. The vertical error bars in the figure represent ±1 standard deviation (SD) of the data. As shown in the figure, the total V2I throughput decreases when the load size increases due to load increases and effective communication resource utilization decreases, resulting in a decrease in system throughput. However, the advantage function is introduced into the algorithm structure proposed in the paper, which alleviates the negative correlation between load size and throughput. The shortest error bars of MADDPG-A2C indicate optimal performance stability. When the load size is 3 × 1060 bytes, the MADDPD-A2C algorithm achieves a performance improvement of 12.5%, 58.1%, 22.7%, and 28.5% over the MADDPG, MADQN, RANDOM, and DDPG baseline algorithms, respectively. Therefore, the total throughput of V2I decreases slightly with the increase in load size of V2V, which has obvious advantages among different algorithms.

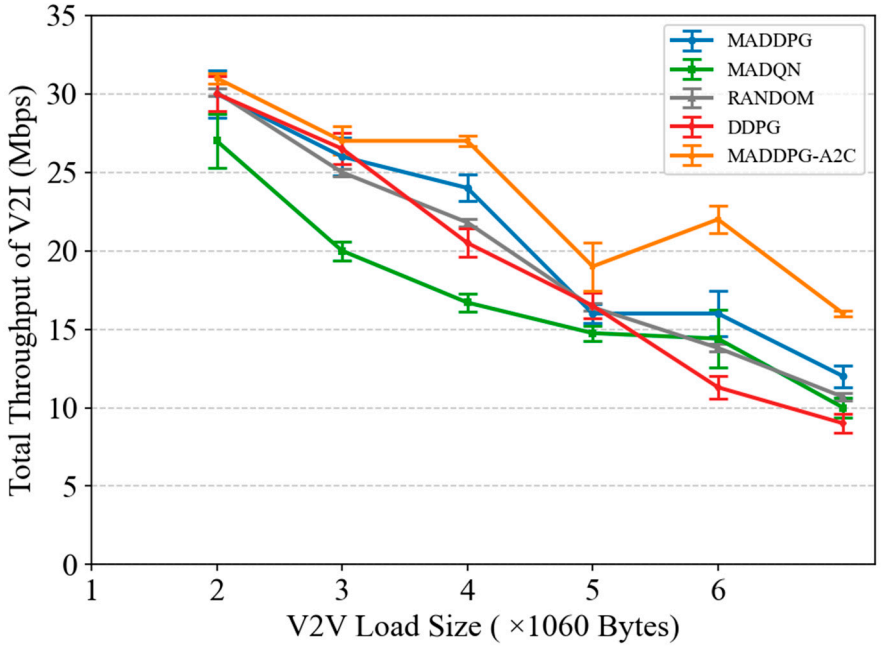
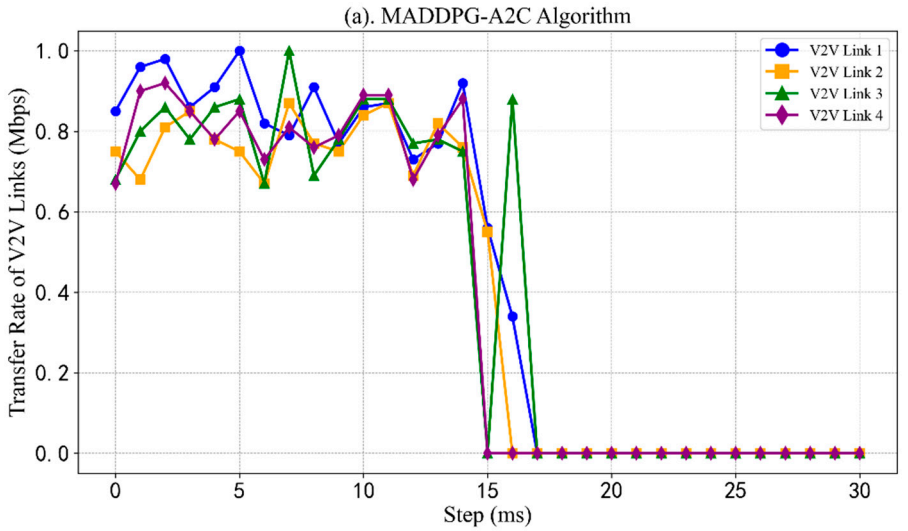
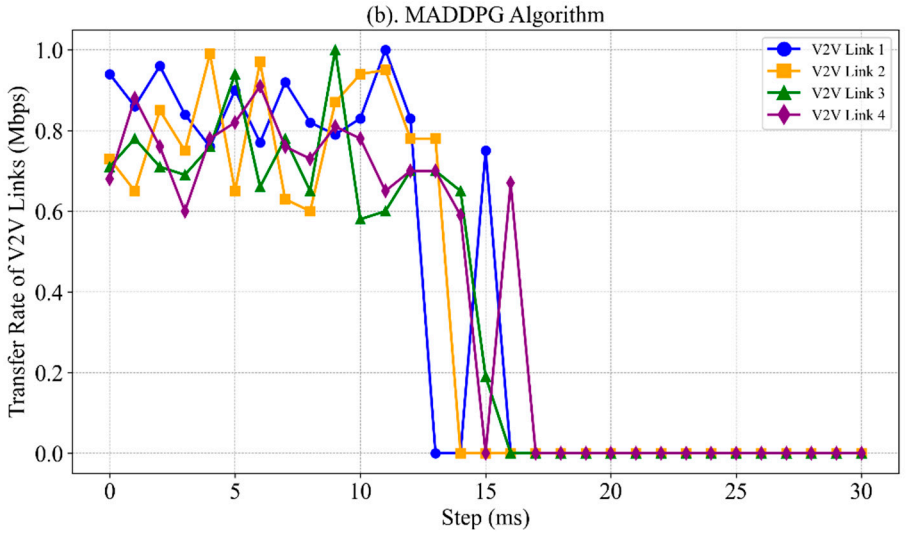


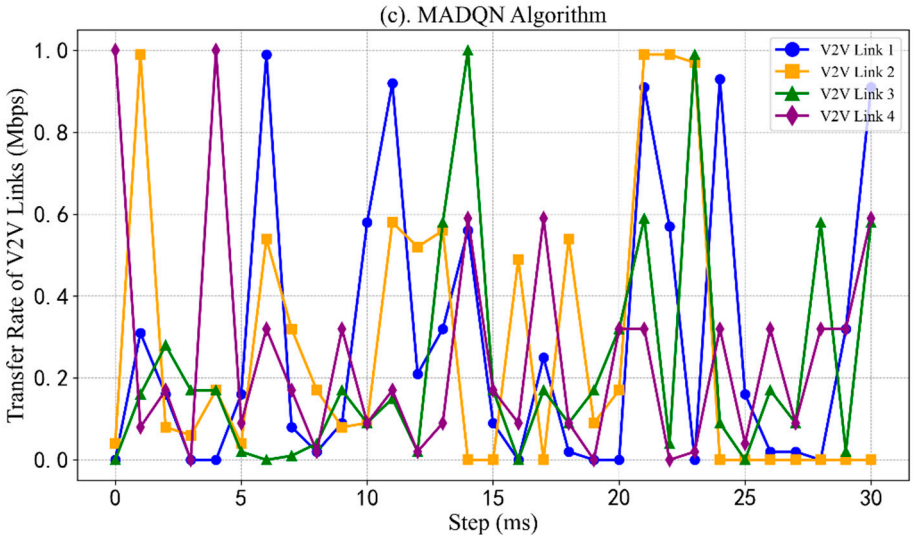
Figure 9. Comparison of the total throughput of V2I links with load size under different algorithms.

Figure 10 shows the transfer rate for the four V2V links under the MADDPG, MADQN, RANDOM, DDPG, and MADDPG-A2C algorithms, respectively. This simulation selects four vehicles (i.e., four vehicle-to-vehicle links) with a load size of 2 × 1060 bytes. It compares the instantaneous rate variations of all V2V links across the five algorithms within 30 ms. It can be observed that all loads of the MADDPG-A2C and MADDPG were successfully transmitted within 17 ms. Compared with the MADDPG algorithm, the instantaneous rate changes of the four links in the MADDPG-A2C algorithm are more stable. The RANDOM algorithm successfully transmitted all link loads within 25 ms. In the MADQN algorithm, only Link 2 was successfully transmitted within 24 ms, while the other links were not completed. In the DDPG algorithm, only Link 3 and Link 4 completed the transmission tasks. This indicates that the MADDPG-A2C algorithm exhibits high efficiency and stability in multi-agent collaborative learning.

https://doi.org/10.3390/electronics15061214







https://doi.org/10.3390/electronics15061214

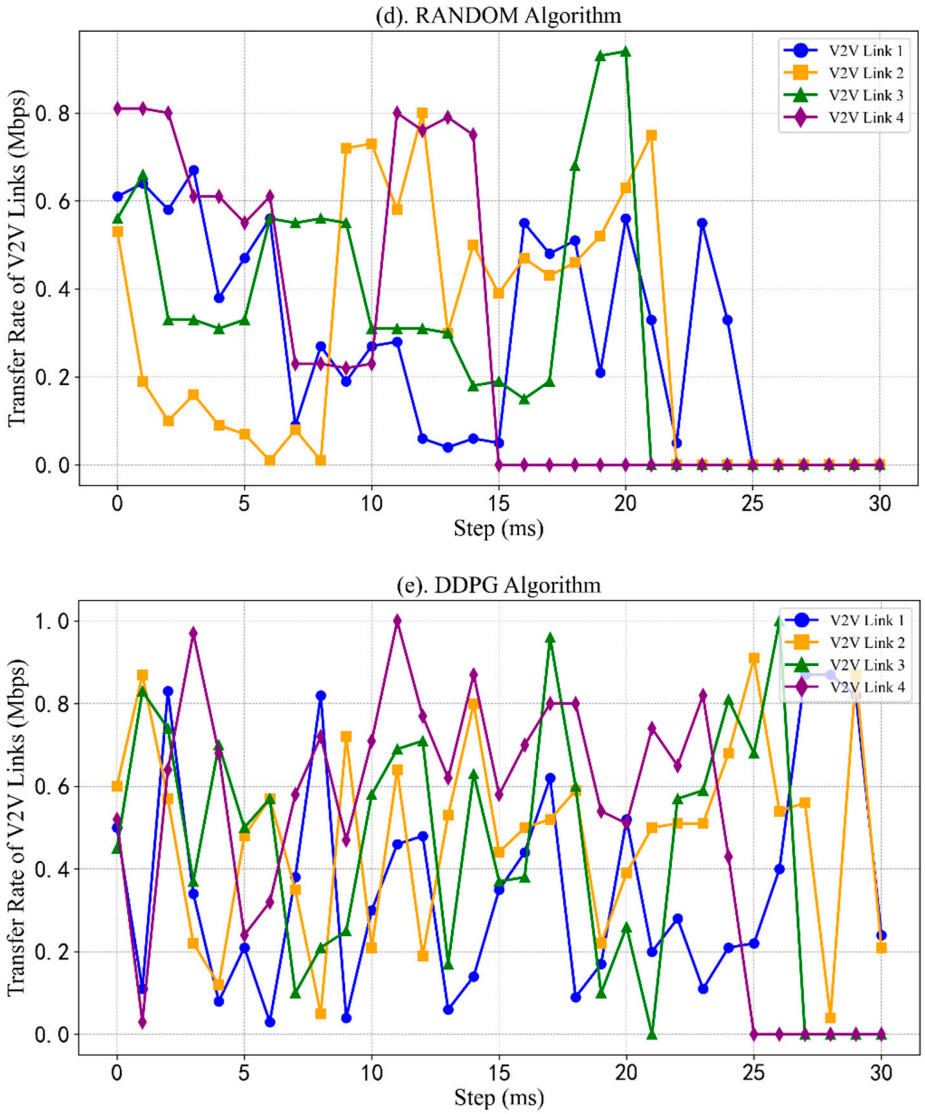
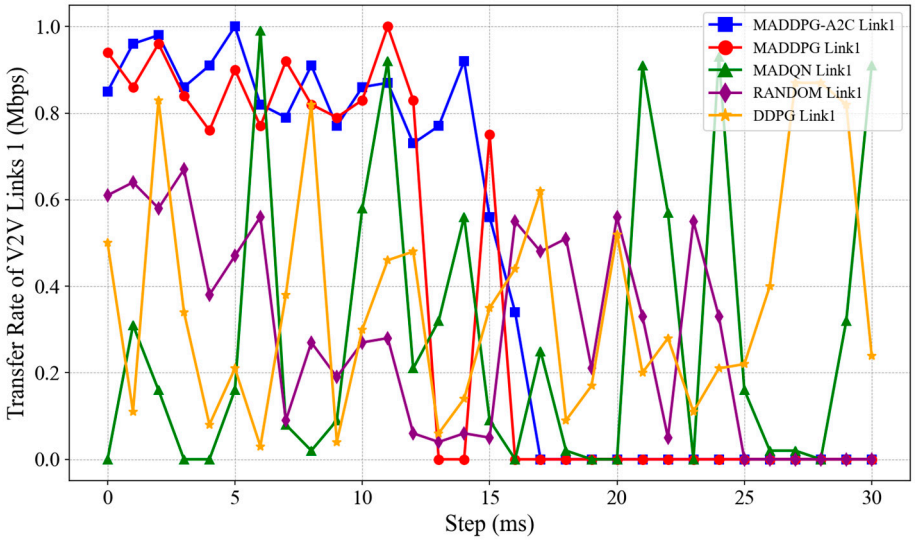
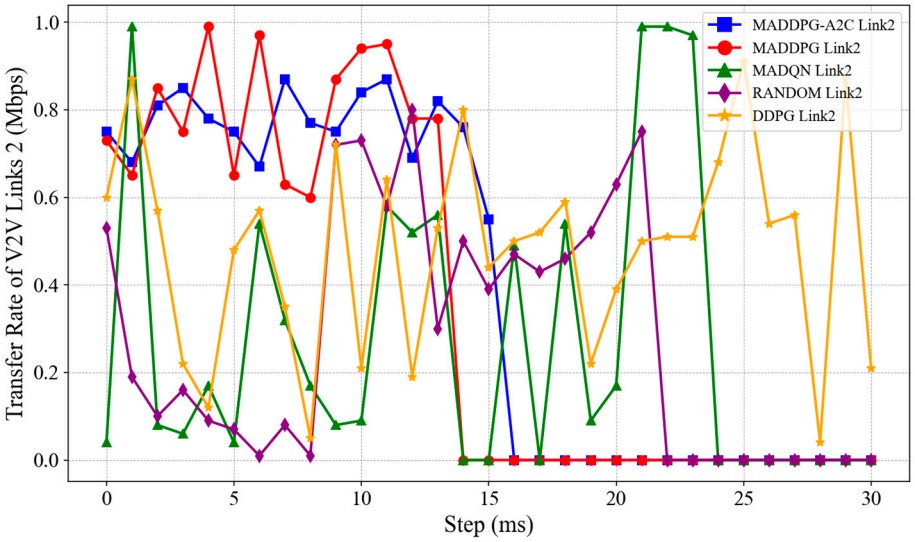


Figure 10. Dynamic transfer rate performance of multi-V2V links with step ((**a**) MADDPG-A2C; (**b**) MADDPG; (**c**) MADQN; (**d**) RANDOM; (**e**) DDPG). (**a**). Transfer rate of four V2V links under the MADDPG-A2C algorithm. (**b**). Transfer rate of four V2V links under the MADDPG algorithm. (**c**). Transfer rate of four V2V links under the MADQN algorithm. (**d**). Transfer rate of four V2V links under the RANDOM algorithm. (**e**). Transfer rate of four V2V links under the DDPG algorithm.

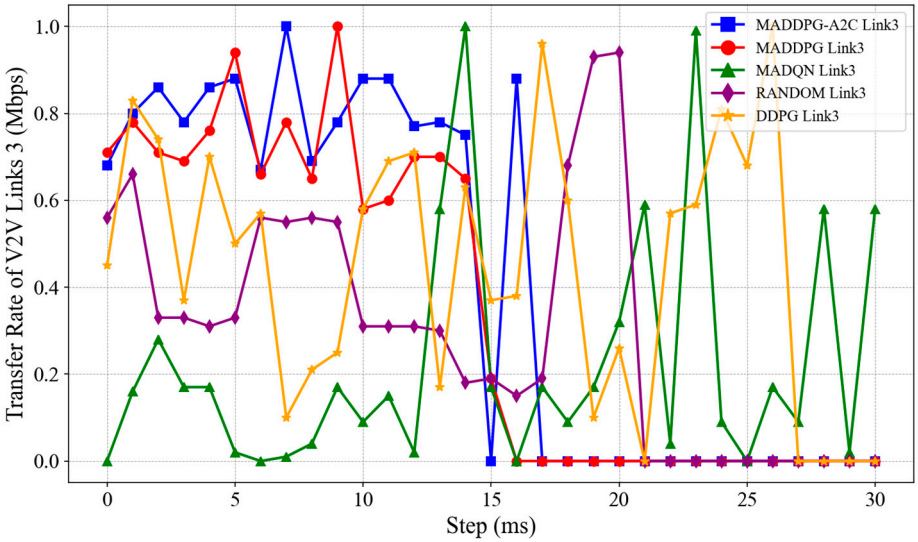
Figure 11 shows the curves of the transfer rate variation with step for four V2V links (Link 1–Link 4) under different algorithms. When the V2V link rate drops to 0, it indicates that the load transmission and delivery task is completed. In V2V Link 1, MADDPG-A2C and MADDPG completed the load transmission task in 17 ms and 16 ms, respectively, which is the shortest time required among the five algorithms. Compared to MADDPG, MADDPG-A2C has a higher average rate and greater stability. In V2V Link 2, Link 3, and Link 4, for the same V2V load transmission task, the MADDPG-A2C algorithm also takes the shortest time, followed by the MADDPG algorithm. This indicates that the agents in the MADDPG-A2C scheme can complete the load transmission task more quickly, and there are no significant fluctuations in each link when completing its own transmission tasks, thereby reducing interference to other links.



(**a**)

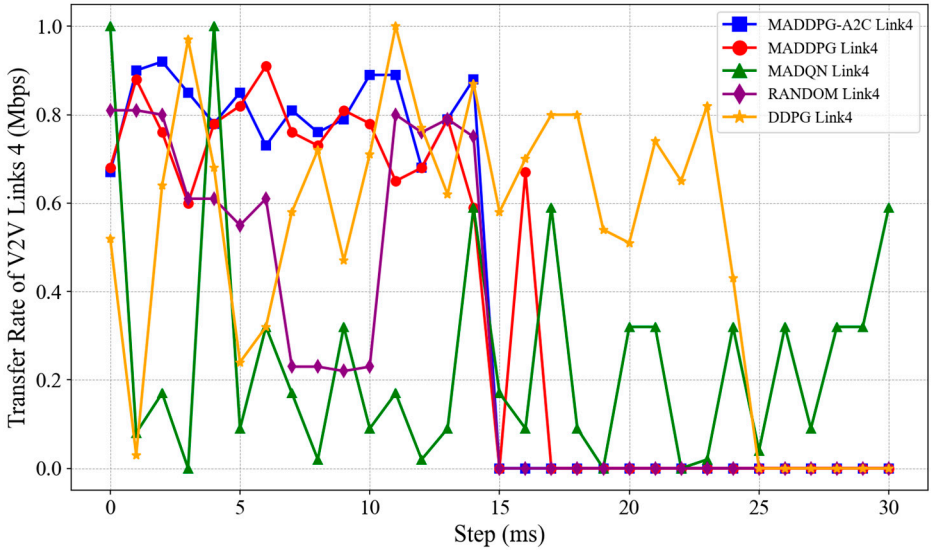


(**b**)



(**c**)

https://doi.org/10.3390/electronics15061214



(**d**)

Figure 11. Comparison of transfer rates of different algorithms under the same V2V link. ((**a**) Link 1; (**b**) Link 2; (**c**) Link 3; (**d**) Link 4). (**a**) Comparison of variations in the transfer rate of V2V Link 1 with step under different algorithms. (**b**) Comparison of variations in the transfer rate of V2V Link 2 with step under different algorithms. (**c**) Comparison of variations in the transfer rate of V2V Link 3 with step under different algorithms. (**d**) Comparison of variations in the transfer rate of V2V Link 4 with step under different algorithms.

To evaluate the robustness of the proposed method under different traffic densities and mobility levels, we conduct additional experiments by varying the number of agents and the vehicle speed range. Specifically, we consider three density settings with 4, 8, and 12 links. The V2V reliability is evaluated by the average access success probability. The channel model uses large-scale fading and shadowing to statistically represent blockage effects, matching real-world obstructed propagation in urban intersection simulations under different building densities.

Figure 12 shows the total V2I throughput versus vehicle velocity under different speed ranges. The speed range gradually increases from [5, 10] m/s to [45, 50] m/s. The results show that for each density setting, the total V2I throughput varies only slightly as the velocity increases from [5, 10] m/s to [45, 50] m/s, indicating that the learned policy is not overly sensitive to mobility. Figure 13 reports the average V2V access success probability versus vehicle velocity. The success probability stays nearly constant across the entire speed range, demonstrating robust reliability under faster channel dynamics and denser spectrum reuse interference.

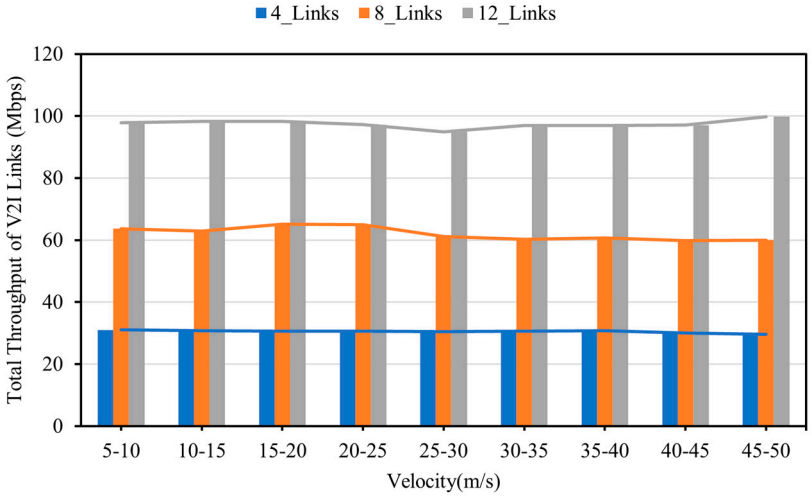


Figure 12. Relationship between vehicle speed and total throughput of V2I under different links.

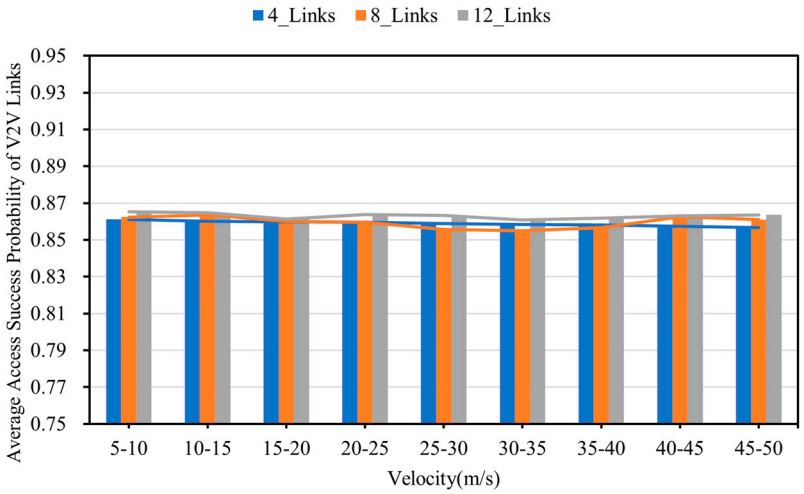


Figure 13. Relationship between speed and average access success probability of V2V under different links.

5.4. Statistical Analysis

To assess the statistical significance of performance differences among the five algorithms (RANDOM, DDPG, MADQN, MADDPG, and MADDPG-A2C) across the three key performance indicators—reward, V2I throughput, and V2V success probability—a statistical test analysis was conducted. The experimental results are shown in Table 4.

Table 4. ANOVA statistical significance test results of five algorithms on key performance metrics ($\alpha = 0.05$).

Methods	Reward	V2I Links' Throughput (Mbps)	V2V Links' Success Probability	MFLOPS (M)
RANDOM	85.09 ± 2.65	29.24 ± 0.87	0.797 ± 0.023	—
DDPG	64.47 ± 1.63	27.43 ± 0.42	0.688 ± 0.015	**0.103**
MADQN	69.65 ± 3.40	27.86 ± 0.62	0.735 ± 0.031	0.207
MADDPG	87.59 ± 1.54	29.80 ± 0.46	0.837 ± 0.021	0.43
MADDPG-A2C	**88.75 ± 1.52**	**30.84 ± 0.40**	**0.851 ± 0.014**	0.516
ANOVA *p*	1.15×10^{-216}	1.19×10^{-288}	8.62×10^{-198}	
ANOVA F	1297.14	285.27	1294.17	

Black bold indicates the best performance metric.

The overall difference test was conducted using one-way analysis of variance (ANOVA, $\alpha = 0.05$). The results showed that the p-values of reward ($p = 1.15 \times 10^{-16}$, F = 1297.14), V2I link throughput ($p = 1.19 \times 10^{-288}$, F = 285.27), and V2V success probability $p = 8.62 \times 10^{-198}$, F = 1294.17) were all much smaller than 0.05. The F-values indicated significant differences between groups, confirming that there were statistically significant overall differences in the three indicators among different algorithms. This indicates that the differences among different algorithms in each indicator are significant, and different methods have statistical distinguishability in performance.

After confirming the overall significant differences among algorithms under each indicator through the ANOVA test, we further conducted post hoc pairwise comparison tests. The comprehensive results showed that MADDPG-A2C performed best in reward, V2I throughput, and V2V success probability, indicating its advantage in improving overall performance.

We further tested the actual computational overhead of different algorithms through experiments, with the results shown in Table 4. The overhead of the original MADDPG is 0.43 MFLOPS, and that of the MADDPG-A2C proposed in this paper is 0.516 MFLOPS. This additional overhead is mainly caused by the forward propagation of the state value network, which is consistent with the results of the theoretical complexity analysis. However, the 0.516 MFLOPS overhead is much lower than the computing power of mainstream in-vehicle edge computing chips and can fully meet the 100 ms level decision delay requirements specified for vehicular network environments in the simulation parameters.

6. Conclusions

This paper studies the dynamic resource allocation problem in vehicular networks, formulates an MADRL objective optimization problem under multi-demand constraints, and proposes the MADDPG-A2C strategy to address the convergence issues faced by traditional reinforcement learning algorithms in high-speed dynamic environments. By designing a multidimensional reward function, it effectively adapts to environmental changes and enhances learning efficiency. The proposed algorithm exhibits excellent scalability and generalization ability, which offers a positive outlook for its research prospects. In the future, a hybrid architecture combining prioritized experience replay and hierarchical experience replay mechanisms will be explored and developed. Meanwhile, the risk of overfitting in high-dynamic scenarios will be considered to balance the training distribution and enhance the model's generalization ability.

Author Contributions: Conceptualization, L.H. and S.X.; methodology, L.H. and W.Z.; software, P.X.; validation, L.H., S.X., and W.Z.; formal analysis, L.H.; investigation, P.X.; resources, W.W.; data curation, B.G.; writing—original draft preparation, S.X.; writing—review and editing, L.H. and W.Z.; visualization, W.Z.; supervision, L.H. and W.Z.; project administration, L.H.; funding acquisition, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China for Young Scholars under Grant No. 62201235 and the High-level Talent Project by the Jinling Institute of Technology (No. jit-b-202302).

Data Availability Statement: The datasets used during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Xiao, Y.; Zhao, J.; Zhang, Q.; Huang, Y.; Quan, H.; Fan, L. MEC-enabled resource allocation in Internet of Vehicles. *Phys. Commun.* **2024**, *65*, 102368.
2. Hua, M.; Chen, D.; Qi, X.; Jiang, K.; Liu, Z.E.; Sun, H.; Zhou, Q.; Xu, H. Multi-Agent Reinforcement Learning for Connected and Automated Vehicles Control: Recent Advancements and Future Prospects. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 16266–16286.
3. Liu, W.; Hua, M.; Deng, Z.; Meng, Z.; Huang, Y.; Hu, C.; Song, S.; Gao, L.; Liu, C.; Shuai, B.; et al. A systematic survey of control techniques and applications in connected and automated vehicles. *IEEE Internet Things J.* **2023**, *10*, 21892–21916.
4. Jin, H.; Seo, J.; Park, J.; Kim, S.C. A Deep Reinforcement Learning-Based Two-Dimensional Resource Allocation Technique for V2I Communications. *IEEE Access* **2023**, *11*, 78867–78878.
5. Ji, B.; Huang, J.; Wang, Y.; Song, K.; Li, C.; Han, C.; Wen, H. Multi-Relay cognitive network with anti-fragile relay communication for intelligent transportation system under aggregated interference. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7736–7745.
6. Chandrasekharan, P.; Jaekel, A. Transmission Power Based Congestion Control Using Q-Learning Algorithm in Vehicular Ad Hoc Networks (VANET). In *Innovations in Mechatronics Engineering III*; Springer: Cham, Switzerland, 2024.
7. Zhang, X.; Peng, M.; Yan, S.; Sun, Y. Deep reinforcement learning based mode selection and resource allocation for cellularV2Xcommunications. *IEEE Internet Things J.* **2020**, *7*, 6380–6391.
8. Xu, X.; Wu, Q.; Fan, P.; Wang, K.; Cheng, N.; Chen, W.; Letaief, K.B. Enhanced velocity-adaptive scheme: Joint fair access and age of information optimization in vehicular networks. *IEEE Trans. Mob. Comput.* **2025**, *25*, 3488–3505.
9. Ji, Y.; Wang, Y.; Zhao, H.; Gui, G.; Gacanin, H.; Sari, H.; Adachi, F. Multi-Agent Reinforcement Learning Resources Allocation Method Using Dueling Double Deep Q-Network in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 13447–13460.
10. Lingayya, S.; Jodumutt, S.B.; Pawar, S.R.; Vylala, A.; Chandrasekaran, S. Correction: Dynamic task offloading for resource allocation and privacy-preserving framework in Kubeedgebased edge computing using machine learning. *Clust. Comput.* **2024**, *27*, 9433.
11. Gyawali, S.; Qian, Y.; Hu, R.Q. Deep Reinforcement Learning Based Dynamic Reputation Policy in 5G Based Vehicular Communication Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6136–6146.
12. Upadhyay, P.; Marriboina, V.; Goyal, S.J.; Kumar, S.; El-Kenawy, E.-S.M.; Ibrahim, A.; Alhussan, A.A.; Khafaga, D.S. An improved deep reinforcement learning routing technique for collision-free VANET. *Rep. Sci.* **2023**, *13*, 21796.
13. Ergün, S. Resource allocation optimization for effective vehicle network communications using multi-agent deep reinforcement learning. *J. Dyn. Games* **2025**, *12*, 134–156.
14. Fang, W.W.; Wang, Y.P.; Zhang, H. Optimization of Communication Resource Allocation in Vehicular Networks Based on Multi-Agent Deep Reinforcement Learning. *J. Beijing Jiaotong Univ.* **2022**, *46*, 64–72.
15. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*; Curran Associates Inc.: New York, NY, USA, 2017; pp. 6382–6393
16. Mnih, V.; Badia, A.P.; Mirza, M. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML'16)*; JMLR.org: Cambridge, MA, USA, 2016; Volume 48, pp. 1928–1937.
17. Tian, J.; Liu, Q.; Zhang, H.; Wu, D. Multiagent Deep-Reinforcement-Learning-Based Resource Allocation for Heterogeneous QoS Guarantees for Vehicular Networks. *IEEE Internet Things J.* **2022**, *9*, 1683–1695.
18. Xu, L.; Chen, W.; Liu, X.; Chen, Y.Y. MADDPG: Multi-agent Deep Deterministic Policy Gradient Algorithm for Formation Elliptical Encirclement and Collision Avoidance. In *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*; Lecture Notes in Electrical Engineering; Springer: Singapore, 2023; Volume 934, pp. 978–981.
19. Wan, K.; Wu, D.; Li, B.; Gao, X.; Hu, Z.; Chen, D. ME-MADDPG: An efficient learning-based motionplanning method for multiple agents in complex environments. *Int. J. Intell. Syst.* **2022**, *37*, 2393–2427.
20. Zhang, H.; Li, J.; Yuan, Q. Edge Service Migration for Vehicular Networks Based onMulti-agent Deep Reinforcement Learning. In *Internet of Vehicles, Proceedings of the Technologies and Services Toward Smart Cities: 6th International Conference, IOV 2019, Kaohsiung, Taiwan, 18–21 November 2019, Proceedings*; Springer: Berlin/Heidelberg, Germany, 2020.
21. Kumar, A.S.; Zhao, L.; Fernando, X. Task Offloading and Resource Allocation in Vehicular Networks: A Lyapunov-Based Deep Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2023**, *72*, 13360–13373.
22. Kang, J.; Chen, J.; Xu, M.; Xiong, Z.; Jiao, Y.; Han, L.; Niyato, D.; Tong, Y.; Xie, S. UAV-Assisted Dynamic Avatar Task Migration for Vehicular Metaverse Services: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 430–445.

23. Peng, H.X.; Shen, X.M. Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 2416–2428.
24. Tang, T.; Li, C.; Liu, F. Collaborative cloud-edge-end task offloading with task dependency based on deep reinforcement learning. *Comput. Commun.* **2023**, *209*, 78–90.
25. Li, Z.; Xu, C.; Wu, Z.R. Deep reinforcement learning based trajectory design and resource allocation for task-aware multi-UAV enabled MEC networks. *Comput. Commun.* **2024**, *213*, 88–98.
26. Fan, C.; Xu, H.; Wang, Q. Multi-agent deep reinforcement learning for trajectory planning in UAVs-assisted mobile edge computing with heterogeneous requirements. *Comput. Netw.* **2024**, *248*, 110469.
27. Nayak, B.P.; Hota, L.; Kumar, A.; Turuk, A.K.; Chong, P.H.J. Autonomous Vehicles: Resource Allocation, Security, and Data Privacy. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 117–131.
28. Ahmed, M.; Liu, J.S.; Muhammad, A.M.M.; Wali, U.K.; Fahd, N.; Al-Wesabi, F.N. MARL based resource allocation scheme leveraging vehicular cloudlet in automotive-industry 5.0. *Comput. Inf. Sci.* **2023**, *35*, 101420.
29. Liu, Q.; Ma, Y. Communication resource allocation method in vehicular networks based on federated multi-agent deep reinforcement learning. *Sci. Rep.* **2025**, *15*, 30866.
30. Cao, T.; Huang, D.; Zhao, J. Application of Improved A2C Algorithm in Traffic Signal Control. *Comput. Eng. Des.* **2024**, *45*, 1713–1719.
31. Wang, Q.; Li, Y.; Zhang, K. Multi-agent Asynchronous Advantage Actor-Critic for Dynamic Task Offloading in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2024**, *23*, 5121–5135.
32. Wiseman, Y. Adapting the H. 264 Standard to the Internet of Vehicles. *Technologies* **2023**, *11*, 103.
33. Chen, J.; Wang, Q.; Liu, Y. AoI-Aware Resource Allocation for C-V2X Networks via Multi-Agent Reinforcement Learning with Attention. In Proceedings of the 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall), Washington, DC, USA, 7–10 October 2024; pp. 1–7.
34. Ji, B.; Dong, B.; Li, D.; Wang, Y.; Yang, L.; Tsimenidis, C.; Menon, V.G. Optimization of Resource Allocation for V2X Security Communication Based on Multi-Agent Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2025**, *74*, 1849–1861.
35. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network. *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception (Release 18): 3GPP TS 36. 101 Version 18.5.0*; 3rd Generation Partnership Project: Sophia, France, 2024.
36. Song, X.Q.; Zhang, W.J.; Lei, L.; Song, T.; Zhao, L. Dynamic resource allocation algorithm for multi-objective joint optimization in Internet of vehicle. *J. Southeast Univ./Dongnan Daxue Xuebao* **2025**, *55*, 266–274.

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