sustainability

MDPI

Article

CVDMARL: A Communication-Enhanced Value Decomposition Multi-Agent Reinforcement Learning Traffic Signal Control Method

Ande Chang [1], Yuting Ji [2], Chunguang Wang [3,*] and Yiming Bie [2]

[1] College of Forensic Sciences, Criminal Investigation Police University of China, Shenyang 110035, China; changande@cipuc.edu.cn
[2] School of Transportation, Jilin University, Changchun 130022, China; jiyt21@126.com (Y.J.); yimingbie@126.com (Y.B.)
[3] State Key Laboratory for Strength and Vibration of Mechanical Structures, School of Aerospace Engineering, Xi'an Jiaotong University, Xi'an 710049, China
* Correspondence: wangchunguang@xjtu.edu.cn

Abstract: Effective traffic signal control (TSC) plays an important role in reducing vehicle emissions and improving the sustainability of the transportation system. Recently, the feasibility of using multi-agent reinforcement learning technology for TSC has been widely verified. However, the process of mapping road network states onto actions has encountered many challenges, due to the limited communication between agents and the partial observability of the traffic environment. To address this problem, this paper proposes a communication-enhancement value decomposition, multi-agent reinforcement learning TSC method (CVDMARL). The model combines two communication methods: implicit and explicit communication, decouples the complex relationships among the multi-signal agents through the centralized-training and decentralized-execution paradigm, and uses a modified deep network to realize the mining and selective transmission of traffic flow features. We compare and analyze CVDMARL with six different baseline methods based on real datasets. The results show that compared to the optimal method MN_Light, among the baseline methods, CVDMARL's queue length during peak hours was reduced by 9.12%, the waiting time was reduced by 7.67%, and the convergence algebra was reduced by 7.97%. While enriching the information content, it also reduces communication overhead and has better control effects, providing a new idea for solving the collaborative control problem of multi-signalized intersections.

Keywords: traffic signal control; deep reinforcement learning; multi-agent reinforcement learning; communication; traffic congestion

Citation: Chang, A.; Ji, Y.; Wang, C.; Bie, Y. CVDMARL: A Communication-Enhanced Value Decomposition Multi-Agent Reinforcement Learning Traffic Signal Control Method. *Sustainability* **2024**, *16*, 2160. https://doi.org/10.3390/su16052160

Academic Editor: Armando Cartenì

Received: 20 January 2024
Revised: 28 February 2024
Accepted: 4 March 2024
Published: 5 March 2024

1. Introduction

With the rapid development of urban motorization, there has been a serious imbalance between traffic demand and supply. Traffic congestion has become a major traffic problem faced by most cities, and its environmental, social, and economic consequences are well documented [1–3]. Traffic signal control (TSC) is one of the effective means by which to solve traffic congestion. It balances the traffic flow in a road network by coordinating the timing scheme of the traffic lights in the control area, so as to reduce the number of stops, delay time, and energy consumption. Promoting the development of traffic control systems is of great significance for giving full play to the traffic benefits of road systems, mitigating environmental pollution, and assisting in the sustainable development of traffic systems.

In recent years, machine learning methods have been widely used in various fields as a new artificial intelligence technology [4–7]. In the reinforcement learning (RL)-based control framework, the traffic signal control system no longer relies on heuristic assumptions and equations, but learns to optimize the signal control strategy through continuous trial and

error through real-time interactions with a road network. Therefore, compared to traditional traffic control methods, RL signal control methods can usually achieve better control effects [8–10]. Early RL-based models solved traffic signal control problems by querying q-tables that recorded the traffic state, actions, and rewards [11,12]. This is easy to implement in environments with relatively simple traffic conditions, but the processing method will occupy a large amount of storage space for relatively complex traffic environments. In this regard, some scholars have chosen to use a q-network to fit a q-table, have applied deep learning (DL) to enhance the ability of RL-based algorithms to cope with complex environments, and have proposed a deep reinforcement learning (DRL) algorithm [13]. Since then, a large number of studies have used DRL algorithms to solve TSC problems, and have achieved good results in practice [14–17].

However, for the signal control of multiple intersections within a certain area (the collaborative control task under a multi-agent system), the partial observability of the traffic environment makes the mapping from the road network state to actions challenging [18]. Communication collaboration between intersections has become an important link that cannot be ignored in effective regional signal control, and the multi-agent reinforcement learning (MARL) algorithm has gradually become one of the most promising methods for large-scale TSC [19–21]. According to the collaborative method, MARL-based control methods can be divided into two types: centralized control methods and distributed control methods.

In the centralized control method, all the signal lights (agents) in the road network are controlled by a unified central controller. Each agent passes the observed local traffic state to the central controller, and the central controller uses a deep network (DNN) to fit the joint action function value, performs action sampling from the corresponding policy network, and then sends it to each agent for execution. The centralized method combines the information from all the agents and implies a communication and coordination mechanism between the agents, so it is easier to obtain the global optimal solution. However, action decisions also need to be made after the traffic state statistics from all of the agents are completed, and the strategy formulation speed is relatively slow. In addition, as the number of agents increases, the action space and state space of the algorithm will grow exponentially [11]. Therefore, in large-scale TSC, the centralized learning paradigm is generally not used in order to avoid the "curse of dimensionality" problem. The distributed control method assumes that the agent is in a stable environmental state and regards the other agents as part of the environment. Each agent optimizes its own strategy in the direction of maximizing the global reward based on its own observations, so the scalability of the distributed control method is relatively good. However, the independent learning method also makes the distributed control method more likely to fall into local optimality [22].

In order to solve this problem, most scholars have incorporated a communication mechanism into the TSC model framework to achieve better control effects. Specifically, communication mechanisms can be mainly divided into two types: "explicit" communication and "implicit" communication [23,24]. The core of explicit communication is to explore how intelligent agents communicate. Among them, the selection of communication objects can be achieved through heuristic frameworks [25,26] and gating mechanisms [27]; the adjustment of the communication content and time is based on DL methods, such as attention mechanisms [18,28], recurrent neural networks [29], and graph neural networks [30–32]. Implicit communication mainly affects the behavioral strategy formulation of the signal agent through value function decomposition and centralized value function [20,23,33–35]. Most implicit communication MARL frameworks use the centralized-training decentralized-execution (CTDE) learning paradigm, which allows the agents to use global (road network) information for centralized learning during the training phase. After the training is completed, each agent can complete the selection of an executable action only through its own observations and local information interactions, which greatly reduces communication overhead while ensuring agent communication cooperation.

In this study, we use the adjustment plan for signal timing as the optimization variables, with the goal of minimizing the average vehicle delay in the road network, and design a multi-agent deep reinforcement learning model considering the communication content based on QMIX [33], namely, CVDMARL. This model combines two communication mechanisms and belongs to the distributed control method under the CTDE paradigm. The contributions of the present study lie in the following:

1. A model that considers the communication content is proposed to solve the regional TSC problem. This model decouples the complex relationships among the multi-signal agents through the CTDE paradigm, and uses a modified DNN network to realize the mining and selective transmission of traffic flow features. It enriches the information content while reducing the communication overhead caused by the increase in information.
2. We design several comparison experiments using traffic data sets from the real world, and examine the advantages of CVDMARL for regional traffic signal control tasks by comparing it to six baseline methods, including the fixed signal control model and five other advanced DRL control models.

The remainder of this study is organized as follows. Section 2 reviews the related research on traffic signal control based on DRL. Section 3 introduces the definition of the problem and the CVDMARL algorithm framework proposed in this paper. The experiments and a performance evaluation are presented in Section 4. Finally, we conclude our work and discuss future prospects in Section 5.

2. Literature Review

2.1. Single-Agent Deep Reinforcement Learning in Traffic Signal Control

The setting of single-agent reinforcement learning mainly consists of two parts: the agent and the environment. The essence of the model is a Markov decision process, called MDP, which is represented by a five-tuple containing the environmental state, action, state transition function, reward, and reward discount coefficient, that is, $G = \langle S, A, P, r, \gamma \rangle$. DRL algorithms based on single agents are mostly applied to traffic control problems at isolated intersections. Researchers usually conduct specific research around two directions of the intersection environment, the feature extraction and model structure improvement. Ma et al. [29] used the historical traffic state as a time series image sequence, mining the spatiotemporal feature information of the traffic flow data based on the combined structure of convolutional neural layers and Long Short-Term Memory (LSTM), and achieved final signal control through the actor–critic framework. Li et al. [36] constructed an adaptive control method for isolated intersection signal control using the signal phase and duration as the actions, and minimizing the average waiting time of vehicles as the goal. Yazdani et al. [37] considered pedestrian travel needs and established a traffic signal adaptive control method based on DRL to minimize delays for all the intersection users (vehicle flow and pedestrian flow). Bouktif et al. [38] considered both discrete and continuous decision making, and used the intersection phase and duration as the optimization variables with which to propose a parameterized, deep q-network architecture. Similarly, Ducrocq et al. [39] proposed a new Deep Q-Network (DQN) model for signal control in a traffic environment where intelligent connected vehicles and ordinary vehicles mix, and adjusted the model architecture and hyperparameters through a partial discrete traffic state coding and delay-based reward functions.

2.2. Multi-Agent Deep Reinforcement Learning in Traffic Signal Control

There are at least two agents in a multi-agent system, and there is usually a certain relationship between the agents, such as cooperation, competition, or both competition and cooperation. The collaborative control problem of multi-signalized intersections is generally a multi-agent control problem under a cooperative relationship, and the traffic lights in the road network are regarded as the intelligent agents. Since the sensors contained in each agent only cover a small part of the overall environment in actual situations, the

signal control model based on multi-agents is usually described as a decentralized, partially observable Markov decision process (DEC-POMDP). This process can be represented by the seven-tuple of $G = \langle S, A, P, r, Z, O, \gamma \rangle$. Among them, $o \in O$ represents the local observation received by the agent, and Z is the observation function. During the training process of the network, each agent learns the control strategy for the traffic signals through a continuous interaction with the environment to achieve the goal of alleviating traffic congestion. However, from the perspective of each agent, the environment is unstable, which is not conducive to convergence. In order to increase the stability of the training, the communication interactions between agents has gradually become a key issue that researchers have paid attention to.

Wang et al. [12] extracted the state representation of the road network environment through the k-nearest neighbor algorithm, and stabilized the model based on spatial discount rewards. Zhu et al. [17] designed a dynamic interaction mechanism based on the attention mechanism to promote information interaction between the agents. On this basis, they used a generalized network to process the joint information and used a ridge regression to update the network parameters. Li et al. [40] proposed a knowledge-sharing deep deterministic policy gradient (DDPG) model, in which each agent has access to the state set collected by all the agents. Yang et al. [41] constructed an RL framework that considers the multi-agent mutual information. They measured the correlation between the input states and output information through the mutual information, and optimized the overall model based on the mutual information. Wu et al. [22] and Chen et al. [42] both used LSTM to alleviate the instability of the local observable states in the environment. On this basis, Wu used a DDPG framework of centralized training and distributed execution to share the environmental parameters, and Chen realized the communication and collaboration between agents based on a value decomposition-based QMIX network.

However, to apply these methods to actual engineering issues, communication limitations such as bandwidth availability are still unavoidable and important issues [43]. The communication network not only brings more useful feature information, but also increases the overall communication overhead of the model to a certain extent. Therefore, how to limit additional communication overhead while maintaining cooperation is still a major challenge facing the road network TSC problem. In response to the above situation, this paper proposes a multi-agent signal control method that combines "explicit-implicit" communication, decouples the complex relationships between the multi-signal agents through the CTDE paradigm, and uses a modified DNN network to realize the mining and selective transmission of traffic flow features. This method enhances the information richness while mitigating the communication overhead stemming from data volume escalation, providing a new idea for solving the collaborative control problem of multi-signalized intersections.

3. Methodology

3.1. Problem Definition

In CVDMARL, a traffic light in a road network is regarded as independent agent $n(n \in \boldsymbol{N} \equiv \{1, \ldots, N\})$, and each agent obtains the state that characterizes the current environment based on the sensor observations within the respective intersection range. The detailed definitions are as follows.

State: For each agent, the traffic state of the intersection consists of the number of vehicles $\{v_l\}_{l=1}^{L_n}$ in each lane, the number of queuing vehicles $\{q_l\}_{l=1}^{L_n}$ in each lane, and the current phase number ρ of the traffic light. Among them, $L_n \in \boldsymbol{L}$ represents the number of entrance lanes at intersection n, and $\boldsymbol{L}$ is the set of lanes at all the intersections in the road network. The global state is the set obtained by splicing the traffic states of each agent.

Action: The phase sequence of a signalized intersection is fixed. Action a is set to the adjustment of the current green light phase, that is, whether to switch the current phase to the next phase: $a = 1$ indicates a switch to the next phase, and $a = 0$ indicates that the current phase is maintained. In addition, we set the maximum and minimum green light

time and the constraint rules for the yellow phase that must be implemented to convert the phase to ensure the reasonable passage of traffic flow.

Reward: Select the delay as the parameter to construct the reward function

$$r^t = \sum_{n=1}^{N} \sum_{l=1}^{L_n} d_{l,n}^t, \tag{1}$$

where r^t represents the reward obtained by agent n taking an action at time t; $d_{l,n}^t$ is the total delay for the l-th lane of intersection n at time t, which is equal to the product of the total number of stopped vehicles and their parking time in the period aa, starting from time t.

3.2. Model Structure

Figure 1 shows the network framework of the CVDMARL model. As shown in the figure, the model consists of three modules: information processing, feature mining, and action value function fitting. Among them, the information processing module simulates the traffic flow in the actual road network through a simulation of urban mobility (SUMO) and obtains the state parameters for subsequent network training. The feature mining module is mainly composed of an improved DNN network. The input information of the network is the initial state $s_s^t \in \mathbb{R}^{N \times \text{ini\_dim}}$ of each agent at time t and the action $a^{t-1} \in \mathbb{R}^N$ of the previous moment (the action of the initial state defaults to 0). The output is the corresponding feature matrix $s^t \in \mathbb{R}^{N \times \text{s\_dim}}$ and the communication matrix $m^t \in \mathbb{R}^{N \times N}$. Based on the communication signals, each agent can selectively communicate with other agents in the road network to obtain the final state characteristic matrix $\vec{s}^t \in \mathbb{R}^{N \times \text{f\_dim}}$.

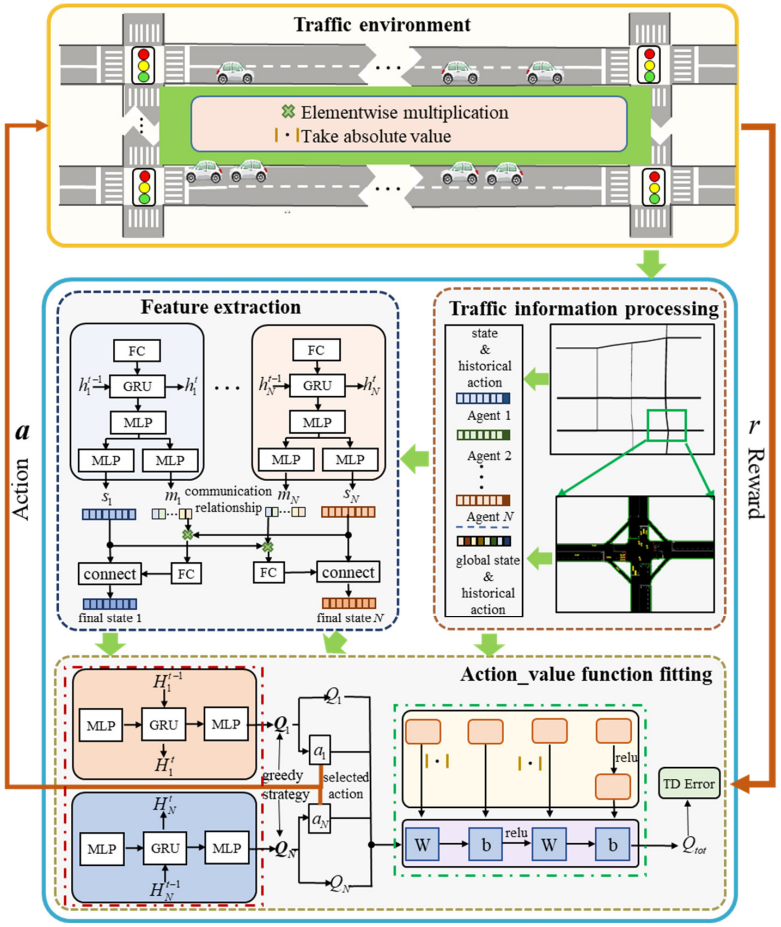

Figure 1. Network framework of CVDMARL.

The action value function fitting network is consistent with the QMIX network. The overall network is mainly composed of the local action value function network (red box network) and the joint action value function network (green box network). The local action value function network belongs to the recurrent neural network (RNN). The input and output of the network are the final feature matrix $\vec{s}^t$ of each agent and the action value function value $Q^t \in \mathbb{R}^{N\times 2}$ of each action, respectively. Based on Q_n^t, each agent uses a greedy strategy to select the optimal action $a^t \in \mathbb{R}^N$, suitable for the current environment, with which to act on the environment. The environment then moves to the next state and returns the reward value r^t under the group of joint actions $\boldsymbol{a^t}$.

The joint action value function network also uses a neural network structure, consisting of a yellow parameter generation network and a purple inference network. The difference is that the weights and biases of the inference network are generated by the parameter generation neural network. At a time t, the parameter generation network accepts the global state S^t and generates the weights and biases. On this basis, the inference network receives the action function values Q^t from all the agents, and assigns the weights and biases generated by the generation network to its own network, thereby inferring the joint action function value Q_{tot}^t. During the training process of the network, based on the joint action value function and reward function of the extracted data, we can calculate the loss function and update the parameters of the network (See Section 3.5 for details).

3.3. Feature Extraction Module

The main framework of the feature extraction module is a modified DNN. Specifically, we use a gated recurrent unit (GRU) to replace a hidden layer in the DNN network to better extract the features. As shown in Equations (2)–(5), the features containing traffic flow information and historical actions are first mapped onto a higher-dimensional vector space to obtain richer semantic information. Then, based on the GRU network, we mine the temporal features in the historical data, and obtain the final feature matrix s^t and communication matrix m^t through two multi-layer perceptron structures with a single hidden layer.

$$f_1^t = W_{\text{f1}}\left[s_{\text{s}}^t, a^{t-1}\right] + b_{\text{f1}}, \tag{2}$$

$$h^t = \text{GRU}\left(f_1^t, h^{t-1}\right), \tag{3}$$

$$f_2^t = \text{ReLU}\left(W_{\text{f22}}\left(W_{\text{f21}}h^t + b_{\text{f21}}\right) + b_{22}\right), \tag{4}$$

$$s^t = W_{\text{f32}}\left(W_{\text{f31}}f_2^t + b_{\text{f31}}\right) + b_{32}, \tag{5}$$

$$m^t = \text{round}\left(\sigma\left(W_{\text{f42}}\left(W_{\text{f41}}h^t + b_{\text{f41}}\right) + b_{42}\right)\right), \tag{6}$$

where $\left[f_1^t, f_2^t\right] \in \mathbb{R}^{N\times \text{ini\_dim}}$ are the intermediate variables in the traffic flow data extraction process; the parameters with W and b as the variables are the trainable weights and biases of the network, respectively; h^t and h^{t-1} are the hidden states at times t and $t-1$, respectively, $\left[h^t, h^{t-1}\right] \in \mathbb{R}^{N\times \text{h\_dim}}$; $\text{ReLU}(\cdot)$ and $\sigma(\cdot)$ are nonlinear activation functions, which can enhance the representation ability and learning ability of the network; and $\text{round}(\cdot)$ is a rounding function that can return the operation result rounded according to the specified number of decimal places.

On this basis, each agent conducts their communication and interaction with each other based on the communication matrix. Taking signal agent n as an example, the communication information corresponding to agent n is located in the n row of the communication matrix m^t, that is, m_n^t. This is an n-dimension bull vector, which is a binary vector composed of 0 and 1. If $m_{n,n'}^t = 1$ (the n' bit in m_n^t), agent n will refer to the environmental information from the

n' agent in order to select an action; otherwise, the environmental information from agent number n' will be ignored. The above process can be expressed by Equations (7) and (8):

$$\begin{aligned} s_{n,1}^t &= m_{n,1}^t s_1^t, \\ s_{n,2}^t &= m_{n,2}^t s_2^t, \\ &\cdots \\ s_{n,N}^t &= m_{n,N}^t s_N^t, \end{aligned} \tag{7}$$

$$\overset{\leftrightarrow}{s}{}_n^t = \text{concatenate}\left(s_{n,1}^t, s_{n,2}^t \ldots, s_{n,N}^t\right), \tag{8}$$

where $\overset{\leftrightarrow}{s}{}_n^t$ is the feature matrix that contains information about the other agents, $\overset{\leftrightarrow}{s}{}_n^t \in \mathbb{R}^{N \times s\_\text{dim}}$.

To facilitate the subsequent calculations, we use a fully connected layer to change the dimension of $\overset{\leftrightarrow}{s}{}_n^t$ and add it to the state vector s_n^t to generate the final state feature $\vec{s}{}_n^t$. Equations (9)–(11) also take agent n as an example to illustrate the flow of information during the generation of the final state features.

$$\overset{\leftrightarrow}{s}{}_n^t = \text{flatten}\left(\overset{\leftrightarrow}{s}{}_n^t\right), \tag{9}$$

$$\tilde{s}_n^t = \sigma\left(\vec{w}\overset{\leftrightarrow}{s}{}_n^t + \vec{b}\right), \tag{10}$$

$$\vec{s}{}_n^t = \text{concatenate}\left(s_n^t, \overset{\leftrightarrow}{s}{}_n^t\right), \tag{11}$$

where s_n^t represents the characteristic state of the n-th agent at time t, corresponding to the n-th row of s^t (calculated by Equation (5)).

3.4. Action Value Function Fitting Module

The composition of the action value function fitting network was introduced in Section 3.1, so this section mainly shows the specific equations corresponding to the module, as well as the detailed meaning of the parameters in it. Equations (12)–(14) show the RNN network, that is, the local value function fitting network. The input for the network is the feature matrix $\overset{\leftrightarrow}{s}{}^t$ of all the signal agents, and the output is the action function value Q^t corresponding to each action in the action set of the signal light agent.

$$q_1^t = \text{ReLU}\left(w_{\text{q12}}\left(w_{\text{q11}}\vec{s}{}^t + b_{\text{q11}}\right) + b_{\text{q12}}\right), \tag{12}$$

$$H^t = \text{GRU}\left(q_1^t, H^{t-1}\right), \tag{13}$$

$$Q^t = w_{\text{q22}}\left(w_{\text{q21}}H^t + b_{\text{q21}}\right) + b_{\text{q22}}, \tag{14}$$

where the parameters with w and b as variables are the trainable weights and biases of the network; the definition of H^t, H^{t-1}, and $\text{ReLU}(\cdot)$ are also consistent with the above, $\left[H^t, H^{t-1}\right] \in \mathbb{R}^{N \times \text{H\_dim}}$.

The calculation of the joint action function value requires the optimal local action function value as the input. In order to implement distributed control under global optimal conditions, the joint action value function and the local value function need to have the

same monotonicity, which means that the action that can maximize the joint action value function should be equivalent to the local optimal action set:

$$\operatorname{argmax}_{\boldsymbol{a}} Q_{\mathrm{tot}}(\boldsymbol{\chi},\boldsymbol{a}) = \begin{pmatrix} \operatorname{argmax}_{a_1} Q_{\mathrm{tot}}(\chi_1, a_1) \\ \vdots \\ \operatorname{argmax}_{a_N} Q_{\mathrm{tot}}(\chi_N, a_N) \end{pmatrix}, \tag{15}$$

where the $\operatorname{argmax}_a(\cdot)$ is used to take the parameters (set) of the function and return the action label corresponding to the maximum value of the action value function; $Q_{\mathrm{tot}}(\boldsymbol{\chi},\boldsymbol{a})$, $Q_{\mathrm{tot}}(\chi_1, a_1), \ldots, Q_{\mathrm{tot}}(\chi_N, a_N)$ are the action value functions of the road network and signal intersection, respectively; and $\boldsymbol{\chi}, \chi_1, \ldots, \chi_N$ are the historical actions of the road network and signal intersection, respectively.

The QMIX network converts the above equation into the constraint condition shown in Equation (16), and satisfies the constraint by restricting the weights w^t_{M1} and w^t_{M2} in the joint action value function network (making their values positive).

$$\frac{\partial Q_{\mathrm{tot}}(\boldsymbol{\chi},\boldsymbol{a})}{\partial Q_n(\chi_n, a_n)} > 0, \ \forall n, \tag{16}$$

$$w^t_{\mathrm{M1}} = \left| w_{\mathrm{m12}}\left(\mathrm{ReLU}\left(w_{m11} s^t + b_{\mathrm{m11}}\right)\right) + b_{\mathrm{m12}} \right|, \tag{17}$$

$$b^t_{\mathrm{M1}} = w_{\mathrm{m2}} s^t + b_{\mathrm{m2}}, \tag{18}$$

$$w^t_{\mathrm{M2}} = \left| w_{\mathrm{m32}}\left(\mathrm{ReLU}\left(w_{\mathrm{m31}} s^t + b_{\mathrm{m31}}\right)\right) + b_{\mathrm{m32}} \right|, \tag{19}$$

$$b^t_{\mathrm{M2}} = w_{\mathrm{m42}}\left(\mathrm{ReLU}\left(w_{\mathrm{m41}} s^t + b_{\mathrm{m41}}\right)\right) + b_{\mathrm{m42}}. \tag{20}$$

In summary, the joint action function value Q^t_{tot} of the road network can be calculated by the following equation:

$$Q^t_{\mathrm{tot}} = w^t_{\mathrm{M2}} \mathrm{ReLU}\left(w^t_{\mathrm{M1}} Q^t + b^t_{\mathrm{M1}}\right) + b^t_{\mathrm{M2}}, \tag{21}$$

where Q^t_{selected} is the action function value under a greedy strategy selection.

3.5. Model Update

The update method of CVDMARL is similar to that of traditional DQN. Both use the TD error to calculate the loss function, and use the backpropagation algorithm to update the network parameters. This process involves two networks: the evaluation network and the target network. The two network structures are identical, as shown in Figure 1, but the input and output information of the two networks are different. The evaluation network takes the features and historical actions in state s as the input, and outputs the actual joint action function value $\check{Q}^{\mathrm{evalutate}}_{\mathrm{tot}}$. The target network takes the features and historical actions of the road network in state s' as the input, and calculates the target (expected) action function value $\widetilde{Q}^{\mathrm{target}}_{\mathrm{tot}}$. The difference between the output content of the two networks constitutes the TD error in state s:

$$\mathrm{TDerror} = \left(R + \gamma \widetilde{Q}^{\mathrm{target}}_{\mathrm{tot}}\right) - \check{Q}^{\mathrm{evalutate}}_{\mathrm{tot}}, \tag{22}$$

$$\widetilde{Q}^{\mathrm{target}}_{\mathrm{tot}} = \max_{a'}\left(Q^{\mathrm{target}}_{\mathrm{tot}}\right), \tag{23}$$

where R is the reward value in state s, and $Q^{\mathrm{target}}_{\mathrm{tot}}$ is the action function value corresponding to all the actions of the target network, $Q^{\mathrm{target}}_{\mathrm{tot}} \in \mathbb{R}^{N\times 2}$.

It can be seen from the above that the calculation of the TD error requires knowing the road network state s at the current moment, the actual joint action taken $\boldsymbol{a}$, the road network state after taking the action, the reward R returned by reaching state s', and the actual joint actions $\widetilde{\boldsymbol{a}}$ taken in the history. Therefore, the calculation of the TD error is not

performed in real time, but is performed after a certain amount of experience has been accumulated. On this basis, the loss function is expressed as follows:

$$loss = \sum_{b=1}^{B} (\text{TDerror}(e_b))^2, \tag{24}$$

where e_b represents the b-th experience in a batch of extracted experiences, and B represents the number of extracted experiences.

In summary, the update process for the CVDMARL framework has the following steps:

1. Initialize the evaluation network, copy its network parameters to the target network, and initialize the experience pool.
2. **Parameters:** The capacity of the experience pool M, the total number of episodes K, the step size of each episode T, and the evaluation-target network update frequency p.
3. **For** k = 1 to K **do:**
4. Initialize the environment, obtain the global state S_{i} of the initial road network, the local observation state s_{i} of each agent, and set the historical action $a_{\mathrm{i}}^{\mathrm{h}}$ of each agent to 0.
5. **For** t = 1 to T **do:**
6. $S^t,\ s^t,\ a^{t-1} \leftarrow S_{\mathrm{i}},\ s_{\mathrm{i}},\ a_{\mathrm{i}}^{\mathrm{h}}$.
7. Taking the local observation state s^t and action a^{t-1} as the input, the feature matrix $\overrightarrow{s}^t$ is obtained based on the evaluation of the feature extraction network.
8. Using $\overrightarrow{s}^t$ as the input, the action function value Q^t in this state is obtained based on the evaluation of the local value function fitting network.
9. Based on the greedy strategy, the action a^t corresponding to the maximum action value is selected with the probability of 1-ε, and randomly selected with the probability of ε.
10. Execute the action, obtain the updated global state S^{t+1}, the local observation s^{t+1}, and the reward r^t.
11. Taking the selected action function value Q_{selected}^t and the global state S^t as The input, the joint action function value Q_{tot}^t is calculated based on the evaluation of the joint action value function network.
12. Store $\left(S^t, s^t, a^{t-1}, a^t, S^{t+1}, s^{t+1}, r^t\right)$ as an experience in the experience pool $\boldsymbol{E}$.
13. **If** len($\boldsymbol{E}$) >= M:
14. Extract B pieces of the experience and update the network parameters.
15. $S^t,\ s^t,\ a^{t-1} \leftarrow S^{t+1},\ s^{t+1},\ a^t$.
16. $t \leftarrow t + 1$.
17. **End for.**
18. If k % p == 0:
19. Copy the parameters of the evaluation network to the target network.
20. **End for.**

4. Experiments

4.1. Experimental Setup

4.1.1. Simulation Settings

Based on the real data sets collected from the actual road network in Fushun City, China, we used SUMO to build a simulation platform and implement model optimization and information interaction through the application program interfaces. The road network simulation environment is shown in Figure 2. The range of the detectors we arranged at each entrance road was 100 m, and the range of the detectors at each exit road was 80 m. For four-way intersections, we used a four-phase signal control scheme of east–west straight, east–west left, south–north straight, and south–north left. For three-way intersections, such as intersections 1, 2, and 7, the signal phase sequence was east–west straight, east/west left, and south/north straight. The duration of the yellow light phase was set to 2 s. We collected the traffic data for the above-mentioned road network during peak hours (6:30–8:30) and off-peak hours (14:00–16:00) for a full week, and summarized

and processed the data into the required input information. The traffic flow distribution for the road network is shown in Table 1 and Figure 3. At the beginning of each simulation, we generated a random number with a value of 0 or 1. If the random number was 0, the simulation platform would load the input information from the peak period, and then generate vehicles second by second based on the built-in code; otherwise, the platform would load the input information from the off-peak periods and generate vehicles second by second based on the built-in code.

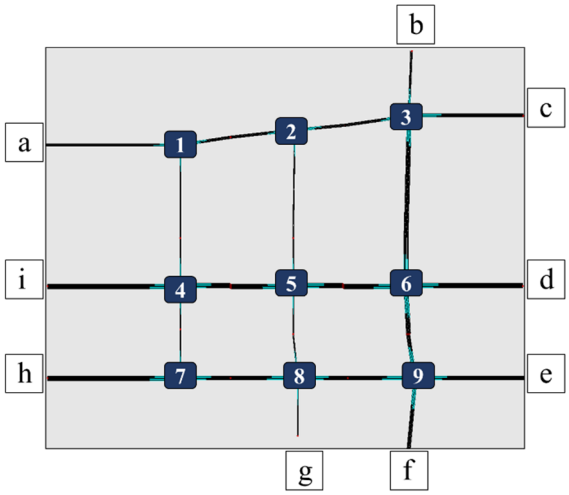

Figure 2. Schematic diagram of road network simulation environment.

Table 1. Traffic volume statistics from the real-world dataset.

Stage	Arrival Rate (veh/300 s)			
	Mean	SD	Max	Min
Off-peak hours	103.55	15.15	139	89
Peak hours	177.20	48.92	255	132

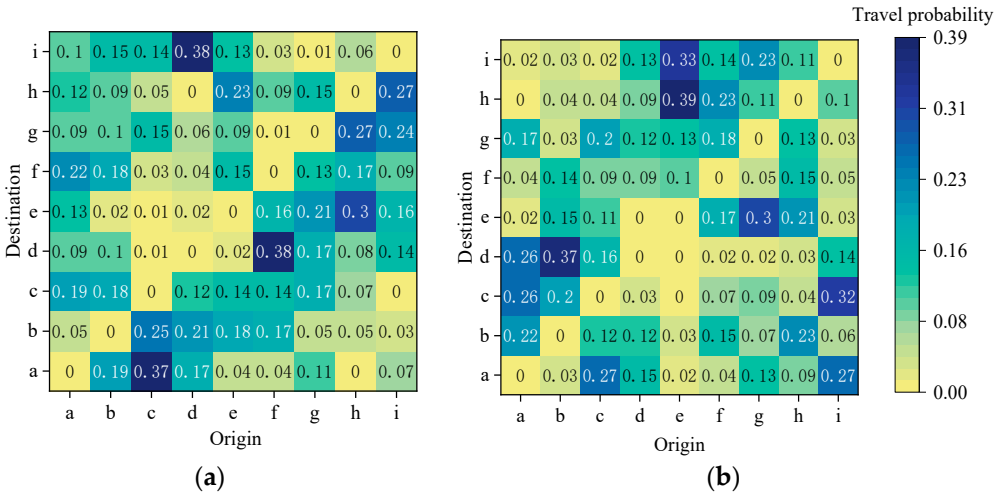

Figure 3. Traffic flow distribution during the test phase. (**a**) Traffic flow distribution during off-peak hours; (**b**) traffic flow distribution during peak hours.

4.1.2. Training Parameters Settings

The duration of each round of training of the CVDMARL model was 3600 s. The parameter settings during the training process are shown in Table 2. The values of these parameters were the results of multiple experiments.

Table 2. Model parameter values.

Parameter	Value	Parameter	Value
B	32	greedy probability ε	0.95–0.01
γ	0.95	initial learning rate lr	0.001
M	1000	s_dim	16
K	200	h_dim	32
T	300	f_dim	32
P	10	H_dim	64

4.1.3. Baseline

(1) FixedTime: A traditional signal control method in which the signal lights run on a fixed timing scheme.

(2) DQN: A centralized control method in which all the intersections are controlled by the same agent. The agent directly fits the joint action value function based on the global environmental state, and then selects the optimal joint action.

(3) IQL: A distributed control method, where each intersection is equipped with independent intelligent agents, and there is no additional information interaction between the intelligent agents. Each agent optimizes its control strategy in the direction of maximizing the global returns based on the global environmental state.

(4) DDQNPER: A communication-free distributed control method that defines the state and reward functions simply and directly, and fits the action value function through Double DQN, with an experience playback function.

(5) QPLEX: Each intersection is controlled by an independent agent, and the action value function of each agent is decomposed into a state value function and an advantage value function. The agent realizes the calculation of the joint action values based on a multi-head attention mechanism, and ensures the consistency of global and local optimality by constraining the value range of the advantage value function. It is a distributed control method with implicit communication.

(6) MN_Light: This method uses a bidirectional LSTM to mine the temporal characteristics of the historical traffic flow status and action information. It is a distributed control method for explicit communication.

4.2. Experimental Results

4.2.1. Comparative Experiment

This section shows the control effects of each baseline method and the CVDMARL model, and further analyzes and discusses the reasons for the above results. As shown in Table 3, we selected three indicators—queue length, waiting time, and travel time—to evaluate the model's effects. Among them, the queue length and waiting time are, respectively, equal to the average queue length and queue time of each intersection entrance lane during the simulation period, and the travel time is the average time required for all the vehicles in the road network to complete their scheduled trip.

Table 3. Comparison of control performance of baseline methods.

Model	Peak			Off Peak		
	Queue Length (m)	Waiting Time (s)	Travel Time (s)	Queue Length (m)	Waiting Time (s)	Travel Time (s)
FixedTime	23.59 (±3.26)	86.35 (±15.56)	218.80 (±35.76)	10.15 (±1.76)	44.35 (±7.04)	173.80 (±33.46)
DQN	16.75 (±1.77)	66.11 (±8.89)	193.13 (±25.76)	7.69 (±1.03)	36.70 (±5.40)	153.90 (±13.26)
IQL	19.98 (±2.11)	75.11 (±11.96)	208.19 (±38.13)	8.07 (±0.70)	38.10 (±8.14)	161.05 (±33.08)
DDQNPER	18.84 (±1.81)	71.11 (±6.64)	199.36 (±17.45)	8.19 (±0.86)	39.06 (±4.83)	162.05 (±18.00)
QPLEX	15.62 (±0.88)	65.04 (±4.47)	195.05 (±11.20)	7.09 (±0.29)	35.65 (±3.02)	156.49 (±16.73)
MN_Light	14.91 (±0.71)	66.83 (±7.82)	193.88 (±19.78)	6.63 (±0.56)	35.02 (±3.26)	157.05 (±23.03)
CMARL	13.55 (±0.67)	61.71 (±6.59)	187.47 (±17.04)	6.27 (±0.41)	34.06 (±3.67)	151.05 (±18.89)

Figure 4 shows the average reward function for each model episode during the training process. Its physical meaning is the total delay for the entire road network (all vehicles) within 12 s (the quotient of the iterative simulation time and the number of episode simulation steps). This indicator is numerically equal to the sum of the reward function values of all the steps in each episode divided by the step size of each episode. Its numerical value reflects the signal control effect of the signal control model on the road network and its changing trend reflects the convergence of the model to a certain extent. As shown in Figure 4, the reward function value of CVDMARL after convergence is significantly lower than that of the other algorithms, and the model convergence speed is relatively fast. The reward function values for DQN and QPLEX after convergence are basically at the same level, but the convergence speed of DQN is significantly higher compared to QPLEX. The reward function value for DDQNPER is slightly higher than the above-mentioned deep reinforcement learning algorithms, and the convergence speed of the model is slow. The reward function value for IQL is relatively high, and the model convergence situation is not advantageous. To further analyze the reasons for the above results, we discuss the model's stability, convergence, and control effects in conjunction with Figure 4.

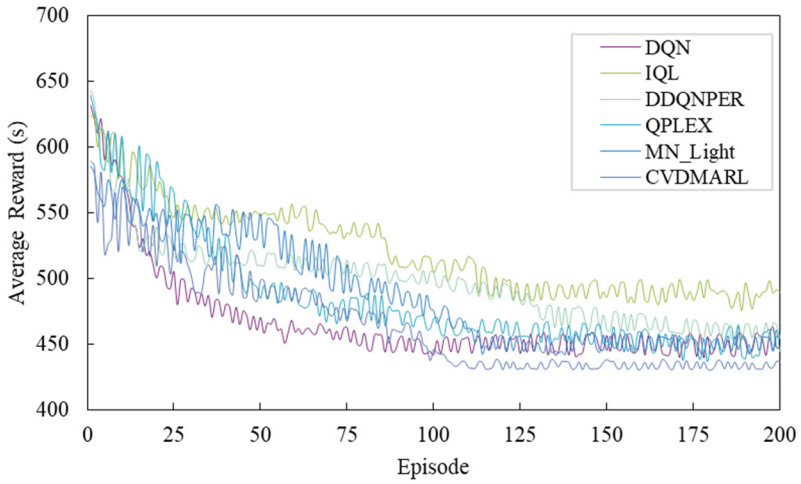

Figure 4. Average reward function change graph during training.

In terms of system stability, some researchers have used a nonlinear analysis and inequality techniques to discuss the stability of system solutions and have combined numerical simulation algorithms to test the correctness and effectiveness of the theoretical results and simulation algorithms through examples [44,45]. In the work related to reinforcement learning, researchers have generally evaluated the control effect and stability of the built model based on the mean, standard deviation, median, and other parameters of multiple test experimental results for different random seed environments after the algorithm's convergence [4,46,47]. Based on the same idea, this paper conducted 10 test experiments with different random seeds after the model's convergence, and the evaluation results for each model are shown in Table 3. Taking the average queue length as an example, during the peak hours, the mean queue length of the CVDMARL model is 13.55, and the standard deviation is 0.67. The mean and variance during the flat peak period are 6.27 and 0.41, respectively, which is significantly lower than those of the other DRL algorithms, and the model's test performance is stable.

As can be seen from the table, various types of DRL algorithms have better control effects than the FixedTime algorithm. In order to further compare and demonstrate the control effects of the various algorithms, we arranged the information in Table 3 into a clustered column chart, as shown in Figure 5. And the average training time and convergence of each model are shown in Table 4. Since the FixedTime model does not have a training process, it is not in Table 4. It can be seen that compared to the basic distributed control methods, such as IQL and DDQNPER, the centralized control method based on

DQN obviously has better control effects, and DQN converged at the 82nd generation, while IQN and DDQNPER converged at the 127th and 131st generations, respectively. This is in line with our inference; that is, a centralized method that collects all the information implies a communication and collaboration mechanism between the agents, and can easily obtain the global optimal solution. However, the basic distributed control method is more likely to fall into local optimality due to the lack of information and interaction between agents. However, the disadvantages of the centralized method are also particularly obvious. The average training time of DQN is as high as 232.4 s, far exceeding that of the other DRL algorithms. As the scale of the road network further expands, the centralized method will occupy more computing space, and the efficiency of the method will decrease significantly.

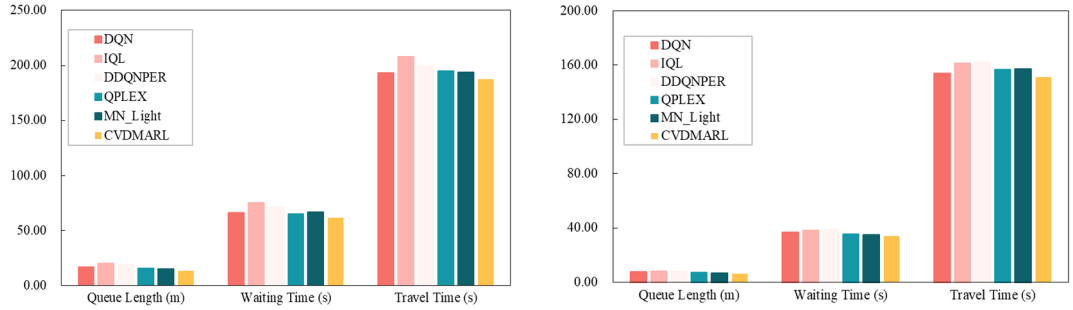

Figure 5. Change diagram of various evaluation indicators of the DRL algorithms relative to the FixedTime method. (**a**) Change diagram of peak hours; (**b**) change diagram of off-peak hours.

Table 4. Average training time and convergence for each model.

Model	DQN	IQL	DDQNPER	QPLEX	MN_Light	CVDMARL
Average Epoch Training Time (s)	232.4	73.6	72.44	83.1	85.7	84.2
Convergent Epoch	82	127	131	102	113	104

QPLEX and MN_Light belong to the distributed control methods of implicit communication and explicit communication, respectively. The former adds the global state information during the training process, and realizes information interconnection by decomposing the global rewards according to their respective contributions. The latter uses bidirectional LSTM to identify the context information to achieve temporal feature extraction in complex environments and enrich the state information that the agent can receive. Compared to the previous two distributed methods, both improved to a certain extent regarding various evaluation indicators: taking the peak hours as an example, compared to DDQNPER, the queue length of QPLEX was reduced by 17.09%, the waiting time was reduced by 8.53%, and the travel time was reduced by 2.16%. MN_Light's queue length was reduced by 20.86%, the waiting time was reduced by 6.01%, and the travel time was reduced by 2.75%. However, due to the increase in network depth, the improved model's control effect also brings more computing overhead. Compared to DDQNPER, the average training time for QMIX and MN_Light increased by 23.13% and 14.16%, respectively.

CVDMARL combines two communication methods: implicit communication and explicit communication. It decouples the complex relationships between multi-signal agents through the CTDE paradigm, and uses an improved DNN network to realize the mining and selective transmission of traffic flow features, with better control effects: compared to the optimal method, MN_Light, among the baseline methods, the queue length of CVDMARL during the peak hours was reduced by 9.12%, the waiting time was reduced by 7.67%, and the travel time was reduced by 3.31%; the queue length during the off-peak hours was reduced by 5.43%, the waiting time was reduced by 2.72%, and

the travel time was reduced by 3.83%. In terms of model complexity, due to the increase in network depth, the average training time for CVDMARL was only reduced by 1.75% compared to MN_Light, but the convergence speed of the network was greatly improved. Compared to MN_Light, the convergence epoch of CVDMARL was reduced by 7.97%. It can be seen that CVDMARL can improve the intersection signal control effect while reducing communication overhead to a certain extent.

4.2.2. Ablation Experiment

To further explore the effectiveness of the proposed feature extraction module, we designed an ablation experiment as shown below. Figure 6 shows the evaluation indicators of QMIX (QMIX is the network framework of CVDMARL after stripping off the feature extraction module), CVDMARL, and the model after removing the GRU module in CVDMARL (Remove_GRU) during the peak and off-peak periods. It can be seen that after removing the feature extraction module, the model's control effect drops significantly. This phenomenon is especially obvious during peak hours. During peak hours, CVDMARL's queue length and queuing time were reduced by 9.73% and 5.64%, respectively, compared to QMIX; while during off-peak hours, compared to QMIX, CVDMARL's queue length and queuing time were reduced by 8.87% and 4.47%, respectively. This is because there are many vehicles in the road network during peak hours, and the spatiotemporal relationship between traffic flows is relatively complex. During this time, relying only on the status information directly obtained by the detector, the agent cannot obtain enough environmental information with which to determine the optimal action. Through a comparison with Remove_GRU, we can also see the importance of the GRU module for improving the model's control effect: taking the average vehicle queuing time as an example, after removing the GRU module, the queuing time of the model increased by 2.44% during peak hours, and increased by 1.42% during off-peak hours.

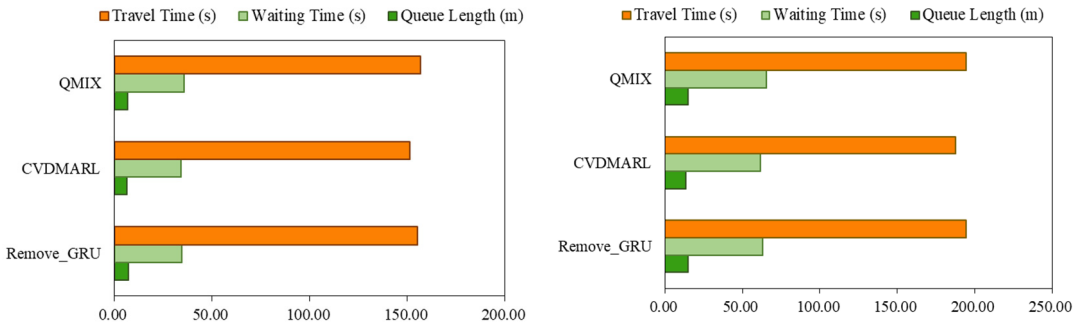

Figure 6. Comparison of QMIX, CVDMARL, and Remove_GRU model effects. (**a**) Comparison of peak hours; (**b**) comparison of off-peak hours.

5. Conclusions

This paper presents a multi-agent deep reinforcement learning model with an emphasis on communication content to solve the signal control problem of road networks. In order to alleviate the instability of model learning caused by local observable states, we use a modified DNN network to excavate and selectively share the nonlinear features in the traffic flow data, enriching the information content and reducing the communication overhead caused by the increase in information. Using real data sets, we conduct a comparative analysis between CVDMARL and six advanced traffic signal control methods, and come to the following conclusions:

(1) CVDMARL can effectively improve the traffic efficiency of a road network, reduce the queuing time and travel time of motor vehicles, and play an important role in alleviating traffic congestion, reducing exhaust emissions, and improving the sustainable development of the transportation system. Compared to the optimal method, MN_Light, among the

baseline methods, CVDMARL's queue length during peak hours was reduced by 9.12%, the average waiting time was reduced by 7.67%, and the average travel time was reduced by 3.31%; the queue length during off-peak hours was reduced by 5.43%, the average waiting time decreased by 2.72%, and the average travel time decreased by 3.83%.

(2) In relatively complex traffic environments, the further extraction of high-dimensional nonlinear features helps the agent to select the optimal actions. After adding the feature extraction module, the model's control effect for QMIX was greatly improved, and the queue length and average waiting time during peak hours were reduced by 9.73% and 5.64%, respectively.

However, this study also has the following limitations:

(1) In the agent design process, the reward function was constructed under the assumption that all the intersections had the same priority, and the impact of the differences between intersections in the road network on the agent's action selection was not fully considered.

(2) We only considered the mining and utilization of the traffic flow's temporal characteristics, and did not further explore the spatial correlation between the road network's intersections.

In future work, we will consider the spatiotemporal differences at road network intersections and design an agent reward function and feature extraction network that are more consistent with the actual situation.

Author Contributions: Conceptualization, A.C.; methodology and validation, Y.J.; writing—original draft preparation, C.W.; writing—review and editing, Y.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the plan project of the Department of Science and Technology, Jilin Province, China [grant number 20230508048RC].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to involvement in other unpublished work.

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

References

1. Zhao, D.; Dai, Y.; Zhang, Z. Computational intelligence in urban traffic signal control: A survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *42*, 485–494. [CrossRef]
2. Kolat, M.; Bécsi, T. Multi-Agent Reinforcement Learning for Highway Platooning. *Electronics* **2023**, *12*, 4963. [CrossRef]
3. Zhang, Z.; Zhang, W.; Liu, Y.; Xiong, G. Mean Field Multi-Agent Reinforcement Learning Method for Area Traffic Signal Control. *Electronics* **2023**, *12*, 4686. [CrossRef]
4. Mao, F.; Li, Z.; Li, L. A comparison of deep reinforcement learning models for isolated traffic signal control. *IEEE Intell. Transp. Syst. Mag.* **2022**, *15*, 160–180. [CrossRef]
5. Osman, M.; He, J.; Mokbal, F.M.M.; Zhu, N.; Qureshi, S. Ml-lgbm: A machine learning model based on light gradient boosting machine for the detection of version number attacks in rpl-based networks. *IEEE Access* **2021**, *9*, 83654–83665. [CrossRef]
6. Jiang, X.; Zhang, J.; Wang, B. Energy-efficient driving for adaptive traffic signal control environment via explainable reinforcement learning. *Appl. Sci.* **2022**, *12*, 5380. [CrossRef]
7. Liu, Y.; Jia, R.; Ye, J.; Qu, X. How machine learning informs ride-hailing services: A survey. *Commun. Transp. Res.* **2022**, *2*, 100075. [CrossRef]
8. Peng, B.; Keskin, M.F.; Kulcsár, B.; Wymeersch, H. Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning. *Commun. Transp. Res.* **2021**, *1*, 100017. [CrossRef]
9. Shi, Y.; Wang, Z.; LaClair, T.J.; Wang, C.; Shao, Y.; Yuan, J. A Novel Deep Reinforcement Learning Approach to Traffic Signal Control with Connected Vehicles. *Appl. Sci.* **2023**, *13*, 2750. [CrossRef]
10. Wang, H.; Zhu, J.; Gu, B. Model-Based Deep Reinforcement Learning with Traffic Inference for Traffic Signal Control. *Appl. Sci.* **2023**, *13*, 4010. [CrossRef]
11. Chu, T.; Wang, J.; Codecà, L.; Li, Z. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1086–1095. [CrossRef]

12. Wang, T.; Cao, J.; Hussain, A. Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103046. [CrossRef]
13. Mannion, P.; Duggan, J.; Howley, E. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic Road Transport Support Systems*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 47–66. [CrossRef]
14. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
15. Haydari, A.; Yılmaz, Y. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Systems.* **2020**, *23*, 11–32. [CrossRef]
16. Liang, X.; Du, X.; Wang, G.; Han, Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1243–1253. [CrossRef]
17. Zhu, R.; Li, L.; Wu, S.; Lv, P.; Li, Y.; Xu, M. Multi-agent broad reinforcement learning for intelligent traffic light control. *Inf. Sci.* **2023**, *619*, 509–525. [CrossRef]
18. Han, Y.; Wang, M.; Leclercq, L. Leveraging reinforcement learning for dynamic traffic control: A survey and challenges for field implementation. *Commun. Transp. Res.* **2023**, *3*, 100104. [CrossRef]
19. Joo, H.; Lim, Y. Intelligent traffic signal phase distribution system using deep Q-network. *Appl. Sci.* **2022**, *12*, 425. [CrossRef]
20. Wan, J.; Wang, C.; Bie, Y. Optimal Traffic Control for a Tandem Intersection With Improved Lane Assignments at Presignals. *IEEE Intell. Transp. Syst. Mag.* **2023**, 2–17. [CrossRef]
21. Liu, Y.; Lyu, C.; Zhang, Y.; Liu, Z.; Yu, W.; Qu, X. DeepTSP: Deep traffic state prediction model based on large-scale empirical data. *Commun. Transp. Res.* **2021**, *1*, 100012. [CrossRef]
22. Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; Wu, D.O. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8243–8256. [CrossRef]
23. Zhuang, H.; Lei, C.; Chen, Y.; Tan, X. Cooperative Decision-Making for Mixed Traffic at an Unsignalized Intersection Based on Multi-Agent Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 5018. [CrossRef]
24. Kővári, B.; Szőke, L.; Bécsi, T.; Aradi, S.; Gáspár, P. Traffic signal control via reinforcement learning for reducing global vehicle emission. *Sustainability* **2021**, *13*, 11254. [CrossRef]
25. Lin, Z.; Gao, K.; Wu, N.; Suganthan, P.N. Scheduling Eight-Phase Urban Traffic Light Problems via Ensemble Meta-Heuristics and Q-Learning Based Local Search. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 14414–14426. [CrossRef]
26. Olayode, I.O.; Tartibu, L.K.; Okwu, M.O.; Severino, A. Comparative traffic flow prediction of a heuristic ANN model and a hybrid ANN-PSO model in the traffic flow modelling of vehicles at a four-way signalized road intersection. *Sustainability* **2021**, *13*, 10704. [CrossRef]
27. Hussain, B.; Afzal, M.K.; Ahmad, S.; Mostafa, A.M. Intelligent traffic flow prediction using optimized GRU model. *IEEE Access* **2021**, *9*, 100736–100746. [CrossRef]
28. Wang, M.; Wu, L.; Li, M.; Wu, D.; Shi, X.; Ma, C. Meta-learning based spatial-temporal graph attention network for traffic signal control. *Knowl. Based Syst.* **2022**, *250*, 109166. [CrossRef]
29. Ma, D.; Zhou, B.; Song, X.; Dai, H. A deep reinforcement learning approach to traffic signal control with temporal traffic pattern mining. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 11789–11800. [CrossRef]
30. Yoon, J.; Ahn, K.; Park, J.; Yeo, H. Transferable traffic signal control: Reinforcement learning with graph centric state representation. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103321. [CrossRef]
31. Yan, L.; Zhu, L.; Song, K.; Yuan, Z.; Yan, Y.; Tang, Y.; Peng, C. Graph cooperation deep reinforcement learning for ecological urban traffic signal control. *Appl. Intell.* **2023**, *53*, 6248–6265. [CrossRef]
32. Xu, M.; Di, Y.; Ding, H.; Zhu, Z.; Chen, X.; Yang, H. AGNP: Network-wide short-term probabilistic traffic speed prediction and imputation. *Commun. Transp. Res.* **2023**, *3*, 100099. [CrossRef]
33. Rashid, T.; Samvelyan, M.; De Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* **2020**, *21*, 7234–7284. [CrossRef]
34. Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; Zhang, C. Qplex: Duplex dueling multi-agent q-learning. *arXiv* **2020**, arXiv:2008.01062. [CrossRef]
35. Ji, J.; Bie, Y.; Wang, L. Optimal electric bus fleet scheduling for a route with charging facility sharing. *Transp. Res. Part C Emerg. Technol.* **2023**, *147*, 104010. [CrossRef]
36. Li, D.; Wu, J.; Xu, M.; Wang, Z.; Hu, K. Adaptive traffic signal control model on intersections based on deep reinforcement learning. *J. Adv. Transp.* **2020**, *2020*, 6505893. [CrossRef]
37. Yazdani, M.; Sarvi, M.; Bagloee, S.A.; Nassir, N.; Price, J.; Parineh, H. Intelligent vehicle pedestrian light (IVPL): A deep reinforcement learning approach for traffic signal control. *Transp. Res. Part C Emerg. Technol.* **2023**, *149*, 103991. [CrossRef]
38. Bouktif, S.; Cheniki, A.; Ouni, A. Traffic signal control using hybrid action space deep reinforcement learning. *Sensors* **2021**, *21*, 2302. [CrossRef]
39. Ducrocq, R.; Farhi, N. Deep reinforcement Q-learning for intelligent traffic signal control with partial detection. *Int. J. Intell. Transp. Syst. Res.* **2023**, *21*, 192–206. [CrossRef]
40. Li, Z.; Yu, H.; Zhang, G.; Dong, S.; Xu, C.-Z. Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103059. [CrossRef]
41. Yang, S. Hierarchical graph multi-agent reinforcement learning for traffic signal control. *Inf. Sci.* **2023**, *634*, 55–72. [CrossRef]

42. Chen, X.; Xiong, G.; Lv, Y.; Chen, Y.; Song, B.; Wang, F.-Y. A Collaborative Communication-Qmix Approach for Large-scale Networked Traffic Signal Control. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 9–22 September 2021; pp. 3450–3455. [CrossRef]
43. Bokade, R.; Jin, X.; Amato, C. Multi-Agent Reinforcement Learning Based on Representational Communication for Large-Scale Traffic Signal Control. *IEEE Access* **2023**, *11*, 47646–47658. [CrossRef]
44. Zhao, K.; Liu, J.; Lv, X. A Unified Approach to Solvability and Stability of Multipoint BVPs for Langevin and Sturm–Liouville Equations with CH–Fractional Derivatives and Impulses via Coincidence Theory. *Fractal Fract.* **2024**, *8*, 111. [CrossRef]
45. Zhao, K. Study on the stability and its simulation algorithm of a nonlinear impulsive ABC-fractional coupled system with a Laplacian operator via F-contractive mapping. *Adv. Contin. Discret. Models* **2024**, *2024*, 5. [CrossRef]
46. Wang, X.; Ke, L.; Qiao, Z.; Chai, X. Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE Trans. Cybern.* **2020**, *51*, 174–187. [CrossRef] [PubMed]
47. Wang, M.; Wu, L.; Li, J.; He, L. Traffic signal control with reinforcement learning based on region-aware cooperative strategy. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 6774–6785. [CrossRef]