

Article

Deep Reinforcement Learning for Ecological and Distributed Urban Traffic Signal Control with Multi-Agent Equilibrium Decision Making

Liping Yan *  and Jing Wang 

Software School, East China Jiaotong University, Nanchang 330013, China; 2021218083500011@ecjtu.edu.cn

* Correspondence: csyan@whu.edu.cn

Abstract: Multi-Agent Reinforcement Learning (MARL) has shown strong advantages in urban multi-intersection traffic signal control, but it also suffers from the problems of non-smooth environment and inter-agent coordination. However, most of the existing research on MARL traffic signal control has focused on designing efficient communication to solve the environment non-smoothness problem, while neglecting the coordination between agents. In order to coordinate among agents, this paper combines MARL and the regional mixed-strategy Nash equilibrium to construct a Deep Convolutional Nash Policy Gradient Traffic Signal Control (DCNPG-TSC) model, which enables agents to perceive the traffic environment in a wider range and achieves effective agent communication and collaboration. Additionally, a Multi-Agent Distributional Nash Policy Gradient (MADNPG) algorithm is proposed in this model, which is the first time the mixed-strategy Nash equilibrium is used for the improvement in the Multi-Agent Deep Deterministic Policy Gradient algorithm traffic signal control strategy to provide the optimal signal phase for each intersection. In addition, the eco-mobility concept is integrated into MARL traffic signal control to reduce pollutant emissions at intersections. Finally, simulation results in synthetic and real-world traffic road networks show that DCNPG-TSC outperforms other state-of-the-art MARL traffic signal control methods in almost all performance metrics, because it can aggregate the information of neighboring agents and optimize the agent's decisions through gaming to find an optimal joint equilibrium strategy for the traffic road network.



Citation: Yan, L.; Wang, J. Deep Reinforcement Learning for Ecological and Distributed Urban Traffic Signal Control with Multi-Agent Equilibrium Decision Making. *Electronics* **2024**, *13*, 1910. <https://doi.org/10.3390/electronics13101910>

Academic Editor: Ping-Feng Pai

Received: 9 April 2024

Revised: 7 May 2024

Accepted: 7 May 2024

Published: 13 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: adaptive traffic signal control; multi-agent reinforcement learning; game theory; mixed-strategy Nash equilibrium; road network

1. Introduction

With the rapid development of urbanization, traffic congestion, traffic accidents, traffic pollution and energy shortages have become important factors restricting sustainable development. Intersection is not only the place where traffic congestion and traffic accidents occur, but also the main place where traffic pollution emissions increase and fuel consumption intensifies [1,2]. Therefore, reasonable traffic signal control methods at intersections can improve the efficiency of urban traffic movement and reduce environmental pollution. However, traditional traffic signal control methods rely heavily on a given traffic flow model, which cannot adapt well to the dynamic traffic flow, and the control efficiency decreases with the increasing traffic flow [3].

The development of information technology and communication technology promotes the traffic signal control method to the intelligent direction. As the core part of modern intelligent transportation, traffic signal control has important practical significance. From the perspective of vehicle road collaboration [4], through optimizing the scheduling of signals, traffic signal control is able to promote the cooperation between vehicles and roads to achieve a more efficient and smooth traffic flow distribution, thus improving the

efficiency of vehicular traffic and reducing traffic congestion, as well as reducing energy consumption and pollution emissions. From the perspective of an intelligent connected vehicle [5], an intelligent traffic signal system adjusts signal timing in real time according to the information exchanged with vehicles and real-time road conditions, optimizing traffic flow to the greatest extent possible, reducing the probability of traffic accidents and improving traffic safety. In addition, traffic signal control [6] can also be combined with intelligent driving technology to achieve safe cooperative driving between vehicles, providing technical support for the development of intelligent transportation.

Urban traffic signal control is essentially a sequential decision-making problem, and Reinforcement Learning (RL) methods are well suited to deal with such problems. RL methods do not need to build a traffic model but rather learn to improve the traffic control scheme by interacting directly with the traffic environment on a continuous basis. A common approach is Single Agent Reinforcement Learning (SARL) [7], where a single agent controls all intersections. At each time step, the agent observes the traffic conditions at all target intersections and selects an action for all intersections; after taking the action, the environment feeds back a reward to the agent. However, this approach is only applicable to traffic signal control in isolated small road networks. Therefore, more researchers have focused on the study of Multi-Agent Reinforcement Learning (MARL) traffic signal control methods [8], where an independent RL agent controls an intersection signal in a large urban road network, and each agent responds to the neighboring traffic state in real time and makes autonomous decisions, which is more in line with the real-world traffic signal control. However, there are new challenges in MARL traffic signal control methods. (1) The state of each agent is affected by the decisions of its neighborhood agents, and there is a non-smooth problem [9]. To solve this problem, interactive communication between agents is expected. However, full interactive communication between agents reduces the value density of collaborative information, while interacting only with neighboring agents limits the scope of collaboration. This is because traffic signal control is a “multilevel hopping” problem, where agents controlling an intersection have to interact on the basis of sensing traffic information representations including multilevel neighbors. However, most of the current MARL traffic signal control methods still cannot achieve effective agent interaction. (2) There is a collaboration problem among agents [10,11]. The relationship among agents in MARL is difficult to handle. Introducing game theory into RL can handle agent interrelationships well, and game theory provides effective solution concepts to analyze the behaviors of multi-agents and describe the learning results of Multi-Agent Systems (MAS) [12]. In addition, through the solution concept of game theory to guide each iteration of the agent, the result can be well converged to the equilibrium point, the convergence speed is faster, and the benefit of each agent is relatively larger. The equilibrium solution of game theory replaces the optimal solution of MARL, and the strategy obtained is also reasonable and effective. However, existing MARL studies only use game theory to empirically update or balance MAS objectives.

At the same time, vehicle actions such as acceleration, deceleration and stopping increase exhaust emissions, and these actions occur with great frequency at intersections, so signal control at intersections directly affects how much exhaust emissions are emitted by vehicles. However, most MARL traffic signal control methods aim to reduce congestion costs by designing more robust algorithms, ignoring the protection of the ecological environment.

To address the above deficiencies, this paper simplifies the multi-intersection traffic environment into an undirected graph model from the perspective of eco-transportation, in which a node represents an intersection agent. In order to accurately capture the spatio-temporal characteristics of each intersection observation in the traffic network, Gate Recurrent Unit (GRU), a convolutional network, is introduced; and multiple GRUs are stacked to achieve the expansion of the collaborative range of the target intersection agent while reducing the communication load of the agents. Secondly, in order to clarify the relationship between the intersection agents and optimize the decision-making process of

the agents, the concept of a mixed-strategy Nash equilibrium solution in game theory is introduced into MARL, and a decision-making process based on the mixed-strategy Nash equilibrium is designed for the agent controlling intersections to optimize and enhance the control capability of the MARL algorithm in the traffic signal control system. Specifically, the contributions of this study are as follows:

1. A Deep Convolutional Nash Policy Gradient Traffic Signal Control (DCNPG-TSC) model framework is constructed, which clarifies the relationship between the agents through the game theory to make the collaboration between the agents more reasonable and expands the sensing domain of the intersection agents through the multi-layer stacked GRU network to make the collaboration between the agents more effective, which greatly improves the real-time response speed of the traffic signals and the traffic efficiency of traffic road network.
2. The Multi-Agent Distributional Nash Policy Gradient (MADNPG) algorithm is designed, which introduces the concept of a mixed-strategy Nash equilibrium solution in the game theory to optimize the agent's decision-making, so that the DCNPG-TSC model can choose an optimal signal phase for each intersection to respond to the real-time changing traffic flow. The DCNPG-TSC model can choose an optimal signal phase for each intersection in respond to real-time changes in the traffic flow and then reduce traffic congestion.
3. Taking eco-transportation as the starting point and incorporating the concept of green transport, the reward value is set to multiple optimization objectives including reducing pollutant gas emission and fuel consumption, thus reducing the pollutant gas emission and energy consumption at intersections.

The rest of the paper is organized as follows. Section 2 presents the existing research work on RL-based traffic signal control. Section 3 establishes the DCNPG-TSC model for the problem and describes the design of Partially Observable Stochastic Game and RL setup in detail. Section 4 describes the specific implementation details of the DCNPG-TSC model, including the design of the agent network structure, the agent decision process, the agent learning process and the MADNPG algorithm. Experimental results and performance analysis are given in Section 5. Section 6 illustrates the conclusions of this paper as well as future perspectives.

2. Related Works

Traditional traffic signal control methods such as fixed-time control, inductive control [13] and adaptive control [14–16] have been widely studied in the past decades. However, with the growth of traffic volume, the complexity of traffic signal control in urban areas is climbing. The traditional traffic signal control methods cannot adapt to the complex and changing traffic flow, and the control effect is unsatisfactory. Thus, scholars have proposed the use of RL methods for traffic signal control.

Early applications of RL to single intersection traffic signal control showed strong advantages. When the road network is extended to multiple intersections, SARL trains a centralized agent to decide the actions at all intersections, which cannot be learned well due to the large dimensionality of the joint action space. Arel et al. [17] proposed a traffic signal control method based on the Q-Learning algorithm and feed-forward neural network, where the outbound agent provides local traffic statistic information to the central agent to collaborate; the central agent learns a value function driven by local and neighboring traffic conditions. However, this method faces scalability problems in deployment and is difficult to be applied in large-scale road networks. To address the above issues, researchers have applied MARL to the traffic signal control problem by training an agent for each intersection and have proposed more complex deep MARL algorithms under the paradigm of centralized training distributed execution. Li et al. [18] proposed the Multi-Agent Deep Deterministic Policy Gradient method to coordinate the traffic signal control, which reduces the waiting time of vehicles by adjusting the phase and duration of traffic lights. Wu et al. [19] proposed a Multi-Agent Recurrent Deep Deterministic Policy Gradient algorithm

based on Deep Deterministic Policy Gradient for how to coordinate traffic controllers at multiple intersections. The algorithm adopts the framework of centralized learning decentralized execution; in centralized learning, agents can estimate other agent strategies during the decision-making process, thus realizing mutual coordination among agents and alleviating the problem of a poor learning effect due to unstable environments; in decentralized execution, each agent can make decisions independently, and the participant network with shared parameters also speeds up the training process and reduces the memory occupation.

The MARL algorithm overcomes the scalability problem in large-scale road networks [20], but it also faces the challenges of how to cooperate, coordinate and share information among agents. The impact of one agent's behavior on the environment is also influenced by the activities of other agents. In addition, individual agents are self-interested, and they want to maximize their returns, and that game relationship leads to locally optimal strategies. Therefore, in traffic signal control applications, optimal decisions need to be made between agents through information sharing and collaboration. However, most of the current research on MARL in traffic signal control focuses on Q-Learning. Abdoos [21] utilizes game theory to coordinate MAS interactive decision making and proposes a two-mode agent structure, in which, in the independent mode, a single agent controls an intersection independently by using Q-Learning; in the cooperative mode, game theory is used to coordinate the cooperation between agents to dynamically control the signal of multiple intersections. Qu [22] designed a distributed control approach to prevent congestion on urban road networks caused by traffic flow demand perturbations by integrating MARL and mixed-strategy Nash equilibrium.

Meanwhile, there are few existing studies on the decision-making process of MARL. MARL makes decisions based on accumulated historical experience, in other words, the ability to make effective decisions is obtained through trial and error. For example, Rasheed et al. [23] applied Deep Q Network (DQN) to traffic signal control to solve the dimensionality problem in traffic congestion and proposed the Mutil-Agent DQN method. Although the concept of game theory is combined in MARL, most of the studies [24,25] use game theory for empirical updating or balancing MAS objectives. Existing studies usually use ϵ -greedy and softmax as strategies for action selection. Decision making in MARL based on mixed-strategy Nash equilibrium, which exhibits a preference for potentially optimal actions in the action selection process and can accelerate MARL convergence. Wu et al. [26] designed two game theory-assisted RL algorithms, Nash Advantage Actor–Critic and Nash Asynchronous Advantage Actor–Critic, by using Nash equilibrium and RL; Wei et al. [27] proposed an algorithm for MARL based on Nash equilibrium and Deep reinforcement learning. The simulation results of the above two methods illustrate that the Nash equilibrium solution of game theory can optimize the agent's decision to achieve adaptive traffic signal control in complex urban road networks, but the above two methods can only be applied to traffic signal control problems with explicit equilibrium solutions, and the scope of application is limited. Meanwhile, there is a gap in the research of RL-based traffic signal control methods on the green emission problem. Li et al. [28] applied a decision tree method in their study, which successfully reduced CO₂ emission and vehicle delay; however, this method is too simple to be applied to complex road networks.

In summary, existing research on traffic signal control based on MARL and game theory focuses on small-scale communication and collaboration with Q-Learning, ignoring the design of the agent decision-making process and failing to consider the green emission problem. For this reason, this paper constructs a DCNPG-TSC model for distributed traffic signal control, which not only considers the communication problem among agents on a large-scale road network, but also takes the collaboration among agents and the agent decision-making process into account, realizes MARL effective communication and collaboration, and substantially improves the control efficiency of traffic signals and alleviates traffic congestion. For the first time, this paper combines mixed-strategy Nash equilibrium and Multi-Agent Deep Deterministic Policy Gradient and proposes the MADNPG

algorithm, which uses mixed-strategy Nash equilibrium to coordinate the relationship between agents and optimize the agent's decision-making process and greatly improves the traffic efficiency of the traffic network; at the same time, this paper incorporates the exhaust emission and fuel consumption into the MARL signal control system, which largely reduces the vehicle exhaust emissions and fuel consumptions and protects the ecological environment of the city.

3. DCNPG-TSC Model for Multi-Intersection Traffic Signal Control

Existing studies on MARL traffic signal control focus on solving the communication between agents, while ignoring the interaction and collaboration between agents. In addition, there are few studies on the combination of MARL and game theory. Therefore, this paper combines MARL and game theory to propose the DCNPG-TSC model. In this section, the general concept of RL control is first introduced from a simple single-agent RL traffic signal control model. Then, the DCNPG-TSC model is proposed to address the shortcomings of the existing research. Finally, the overall structure of the DCNPG-TSC model and the RL settings in the model are described in detail.

3.1. DCNPG-TSC Model Framework

Urban traffic signal control is a typical MAS, where each intersection is abstracted as a single independent agent. Figure 1a illustrates a real road network, which can be simplified to a graph model as in Figure 1b. In the figure, nodes circled in red represent agents; nodes circled in blue denote the set of neighboring nodes that are directly adjacent to the agent; and the purple circle describes a portion of the agent's perceptual domain.

Due to the large scale of urban road network and the complex and changing environment, it is costly and unsatisfactory to consider all intersections at the same time. Therefore, this paper defines the multi-intersection traffic signal control problem as a RL task of a multi-agent game. The whole process is modeled as a Partially Observable Stochastic Game, which consists of a multi-agent game, and described as an eight-tuple, $G = \{N, S, \{A_i\}, \mathcal{P}, \Omega, \{\mathcal{O}_i\}, \{\mathcal{R}_i\}, \gamma\}$, where N denotes the set of finite intersection agents in the traffic signal control region, S represents the set of observed states of all intersection agents in the region, A_i represents the set of actions that can be taken by intersection agent i , $\mathcal{P}$ represents the state transfer function, Ω represents the observation function, $\mathcal{O}_i$ represents the set of observations of intersection agent i , $\mathcal{R}_i$ represents the reward function of intersection agent i , and γ denotes the reward discount factor. Figure 1c shows the process of agent i controlling a traffic signal, in any moment t , agent i observes the real-time traffic flow and obtains a local observation o_i^t ; then, based on the existing observed state o_i^t , it takes a corresponding action a_i^t to obtain an instantaneous reward r_i^t , and the environment transfers to the next moment's state o_i^{t+1} ; agent i observes the impact of this action on the intersection traffic, evaluates the effect of the action just chosen and optimizes the strategy until it converges to the optimal "state and strategy".

In the framework of a Partially Observable Stochastic Game, the agent is in a multi-agent environment, and the agent's decisions are not only affected by the environment but also by the decisions of other agents. Meanwhile, since multi-intersection traffic signal control involves interactions among multiple agents, problems involving interactions among multiple agents usually need to deal with the relationships among multiple agents. In addition, game theory is a powerful tool for portraying the interrelationships among multiple agents, which can deal with the interrelationships among agents well. In order to optimize the strategies acquired by agents and improve the learning efficiency, this paper introduces game theory to clarify the influence of other agents on the current agent decision. In addition, taking all agents into account, the MARL traffic signal control problem requires a high cost, and the RL algorithm that considers all agents at the same time does not substantially improve the control efficiency of the traffic signal control system. The handling of the relationship between agents at intersections is the key to achieving

collaborative MARL traffic signal control. In order to achieve collaborative control of traffic light signals at multiple intersections, this paper adopts game theory to clarify the relationships among neighborhood agents, and at the same time, in order to improve the access efficiency of the urban road network and to reduce the emission of pollutants at the intersections, this paper proposes a distributed MARL framework based on the game equilibrium—the DCNPG-TSC model framework. Figure 2 shows the overall structure of the DCNPG-TSC model.

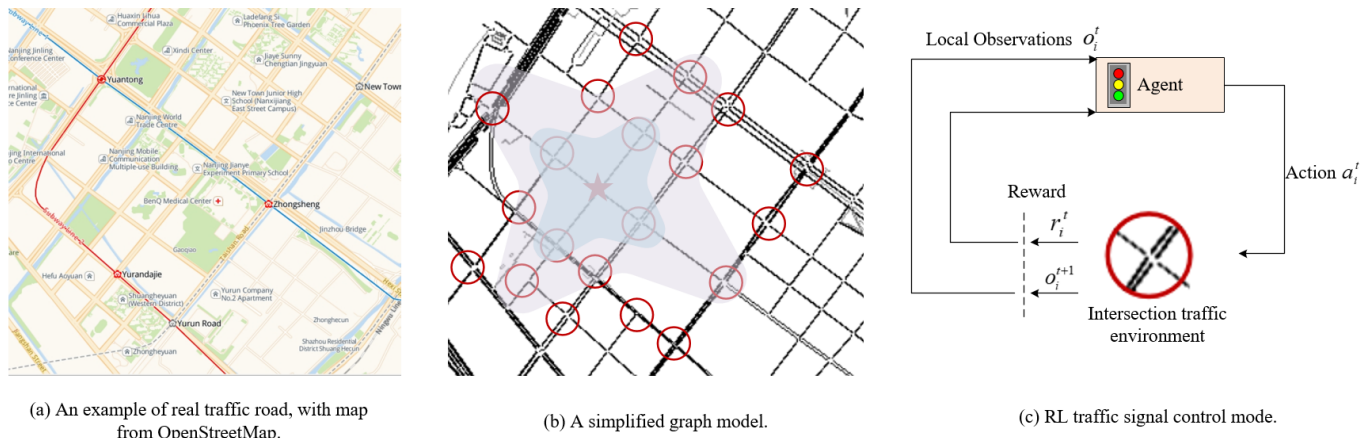


Figure 1. RL traffic signal control framework.

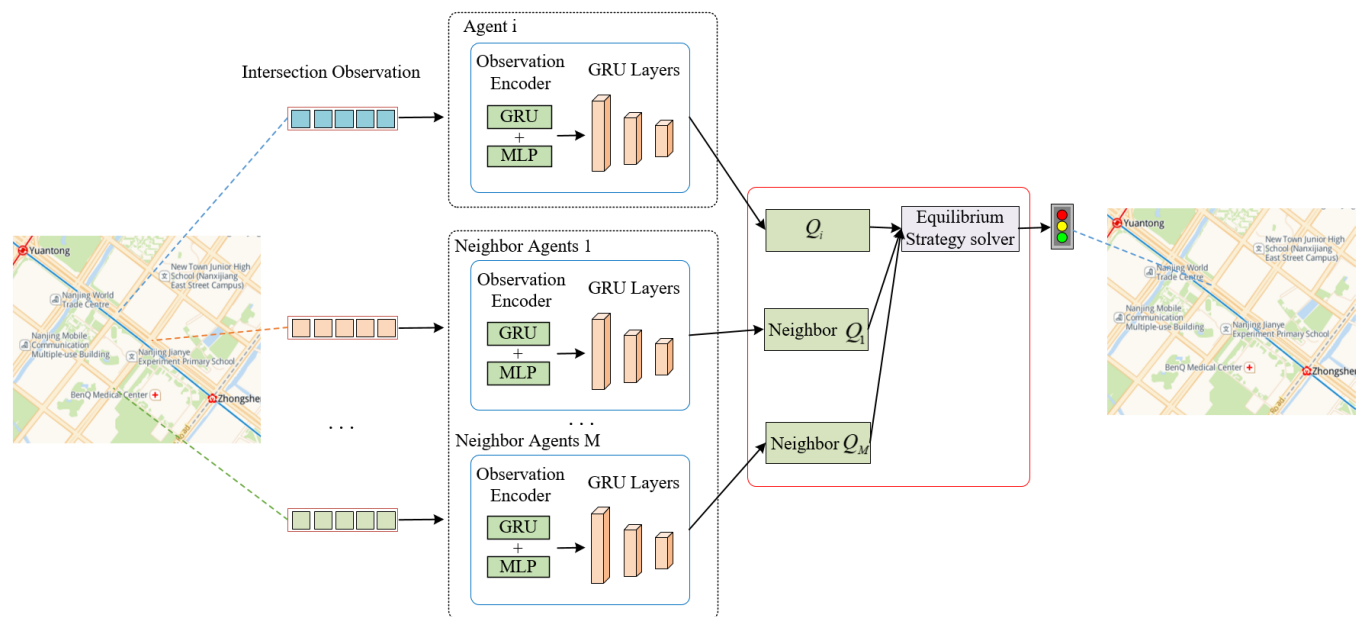


Figure 2. DCNPG-TSC model framework. In this, all the road network maps are obtained from OpenStreetMap.

In this framework, each signal control intersection is abstracted as an agent, and each agent completes the signal control scheme formulation for one intersection with multiple distributed agents learning in parallel. In each time step, all traffic signal controllers (agents) collect traffic state data from each intersection. Then, agents make action decisions based on the observed traffic state, information exchanged with neighboring agents and the game mechanism and select an optimal action for each agent and deliver it to the corresponding intersection for implementation. Subsequently, the environment feeds back an immediate reward and moves to the next state. Finally, agents evaluate the actions they have just taken

and optimize their strategies. Agents thus learn by interacting with the environment to obtain optimal equilibrium strategies in response to real-time changes in the traffic flow on the road network.

In order to achieve global coordinated control, the network design of the DCNPG-TSC model is designed by stacking multiple convolutional layers to expand the sensory domain of the agent, and the training of the agent is based on the observation information and actions of the neighborhood agents to find the optimal equilibrium strategy. The agent network structure is similar to the brain of the DCNPG-TSC model, while the agent learning process makes the DCNPG-TSC model show “intelligence”, and the combination of the two forms an “Agent” that can respond to the traffic environment in real time. These two parts will be elaborated in Section 4. The part framed by the blue line in Figure 2 is the network structure of the agent, and the red line is the agent learning process. MLP is a multi-layer perception machine, GRU stands for Gate Recurrent Unit, and the combination of the two realizes the encoding of traffic observation states; the stacked network of multiple GRUs realizes the extraction of potential traffic features in the gradually increasing perception domain; and Q is the action value function of RL.

3.2. RL Traffic Settings

As the core part of the DCNPG-TSC model is MARL, the definition of the state space, action space and reward function as the three basic elements of MARL is a key part of the DCNPG-TSC model. The definition of the traffic state, action space and reward function are described in detail below.

(1) Intersection observation information o_i^t indicates that at the moment t , the real-time traffic environment information recorded by the intersection traffic cameras and sensors constitutes the local part of the observable state vector of the intersection agent ($i \in \mathcal{N} := \{1, 2, \dots, N\}$), which contains q_i^t, d_i^t . q_i^t indicates the length of vehicles in the queue on all entrance lanes of the intersection; and d_i^t indicates the average delay to vehicles on the entrance lanes of the intersection.

(2) Action a_i^t represents the set of phases taken at an intersection (agent i) at the moment t , which is the set of all traffic signal phases combined by the traffic signal controller. For example, in a synthetic road network consisting of standard “cross-shaped” intersections with 14 lanes, each direction of traffic at the intersection has a separate left-turn lane, and the red and green signals of the traffic signal controllers at each intersection are combined into four phases to form a set of actions corresponding to the intersection, as shown in Figure 3. Among them, the S-N straight and right-turn phases represent the first phase; the S-N left-turn phase represents the second phase; the E-W straight and right-turn phases represent the third phase; and the E-W left-turn phase represents the fourth phase. Since each intersection in a four-phase synthetic road network is identical, each intersection agent in a four-phase synthetic road network has the same set of actions $A = \{a_1, a_2, a_3, a_4\}$. The number and order of phases varies from intersection to intersection in a realistic road network, and the set of actions of the intersection agent is determined by the actual situation. Table 1 shows the correspondence between intersection agent actions, phases, action codes and traffic signal sequences in synthetic road networks. For example, phase I corresponds to action a_1 that is encoded as “0”, and the traffic signal sequence is “GGrrrrrGGrrrrr”. Wherein “G” represents a green signal and “r” represents a red signal. In order to facilitate understanding, “GGrrrrrGGrrrrr” can be regarded as a traffic signal sequence composed of [3, 4, 3, 4] characters. The first three characters “GGr” indicate the three lanes of the north entrance, which indicates that the right turn and straight lanes are green and the left turn lane is red; “rrrr” indicates the four lanes of the east entrance, where the right turn lane, the two straight ahead lanes and the left turn lane are all red. The next three characters and four characters represent the traffic light status of the lanes at the south and west entrances, respectively.

(3) Reward r_i^t indicates the immediate reward that the environment feeds back to the intersection agent i at the moment t . The urban road network traffic signal control problem

has a variety of different control objectives, such as reducing emissions, reducing queue lengths, etc. Different reward functions need to be set for different control objectives. In this paper, the optimization objective is positioned to improve the traffic efficiency of the traffic road network while reducing the emissions as much as possible, so the reward is set to be the sum of the weights of the traffic metrics such as queue length, vehicle waiting time, fuel consumption and tailpipe emissions. In order to investigate the performance of this modeling framework in fuel consumption and tailpipe emissions, this paper adopts the intersection evaluation metrics generator tool that comes with SUMO (Simulation of Urban Mobility, SUMO) to estimate the fuel consumption and the emissions of pollutant gases CO and NOx. Therefore, the reward function is defined as

$$r_i^t = - \left(c_1 \times \sum_{j \in I_{-i}} \left(queue_{l_{ji}}^{t+\Delta t} + wait_{l_{ji}}^{t+\Delta t} \right) + c_2 \times (CO[i] + NOx[i]) \right) \quad (1)$$

where c_1, c_2 is the weight coefficient, I_{-i} is the set consisting of agent i 's neighboring agents, $queue_{l_{ji}}^{t+\Delta t}$ is the queue length measured along the inlet lane l_{ji} at the time t , and $wait_{l_{ji}}^{t+\Delta t}$ is the sum of the vehicle waiting time in the inlet lane l_{ji} at the time t . It is worth noting that the reward is lagged, so both $queue_{l_{ji}}^{t+\Delta t}$ and $wait_{l_{ji}}^{t+\Delta t}$ are measured at time $t + \Delta t$.

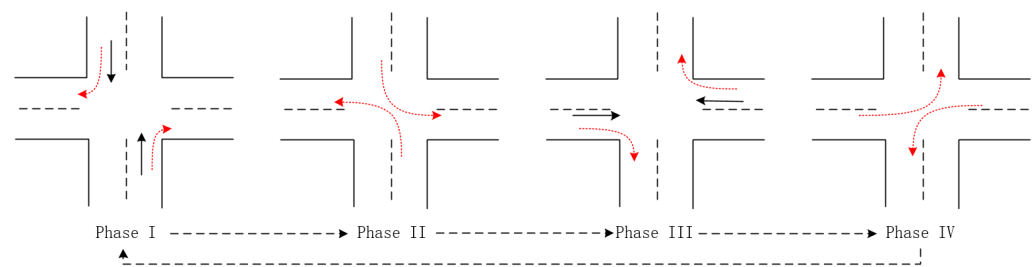


Figure 3. Schematic diagram of intersection agents' phases in a synthetic road network with four phases.

Table 1. The correspondence relationships between agent actions, phases, action codes and traffic signal sequences in the synthetic road network.

Phases	I	II	III	IV
Action	a_1	a_2	a_3	a_4
Encoder in agent	0	1	2	3
Traffic signal sequence	GGrrrrrGGrrrrr	rrGrrrrrrGrrrr	rrrGGGrrrrGGGr	rrrrrrGrrrrrrG

4. MADNPG Algorithm for Multi-Intersection Traffic Signal Control

The core parts of the DCNPG-TSC model in Figure 2 are the design of the agent network structure and the design of the agent learning process. In the agent network structure, the traffic state is firstly passed into the Observation Encoder composed of the Fully Connected network and the GRU network for encoding. Furthermore, it is then passed into the GRU network to extract the implicit features, and multiple GRUs are stacked to increase the agent perception domain, so that the collaboration range is no longer limited to the directly adjacent intersections. In the agent's learning process, in order to find the mixed-strategy Nash equilibrium solution, this paper uses the Boltzmann exploration to predict the strategies of neighbor agents; agent i makes the action decision based on the local observation and the local mixed-strategy Nash equilibrium and proposes the MADNPG algorithm. The MADNPG algorithm is used to train the DCNPG-TSC model to find the optimal equilibrium policy of the multi-intersection traffic signal control system, thus realizing the coordinated and efficient control of multi-intersection traffic signals.

Specifically, the MADNPG algorithm trains the agents in the DCNPG-TSC model to find the optimal equilibrium policy, which determines the actions that the agents should take at the moment t . Section 4.1 elaborates the overall design of the network structure and the specific design of the agent network structure. Section 4.2 introduces the decision-making process based on the mixed-strategy Nash equilibrium, which is the solution process of the Nash policy. Section 4.3 demonstrates the agent learning process, and Section 4.4 gives the specific MADNPG algorithm implementation process.

4.1. Neural Network Structure Design

Considering that traffic flow is complex spatio-temporal data, in order to prevent the intersection agent from only knowing the current state and causing the Markov Decision Process to become non-stationary, this paper uses GRUs that can save hidden states and memorize short-term histories to design the network structure of the DCNPG-TSC model. GRU is a Recurrent Neural Network proposed to solve the problems of long-term memory and gradient in backpropagation, which is a variant of Long Short-Term Memory with a simpler structure than Long Short-Term Memory. Compared with Long Short-Term Memory, it is easier to train using GRU, which can improve the training efficiency to a great extent and can achieve considerable results. Figure 4a shows the overall design diagram of the network structure in this paper, in which the queue state, wait state, CO and NOx, as well as the information of neighboring agents, are firstly inputted into the Fully Connected layer for processing. Then, all hidden neurons are combined and inputted into the GRU layer for processing, and at the same time, multiple convolutional layers are stacked to make the extracted regional environmental embedded in the regional environmental features more concentrated. Finally, the output layer performs a softmax operation on the actor and linear operation on the critic.

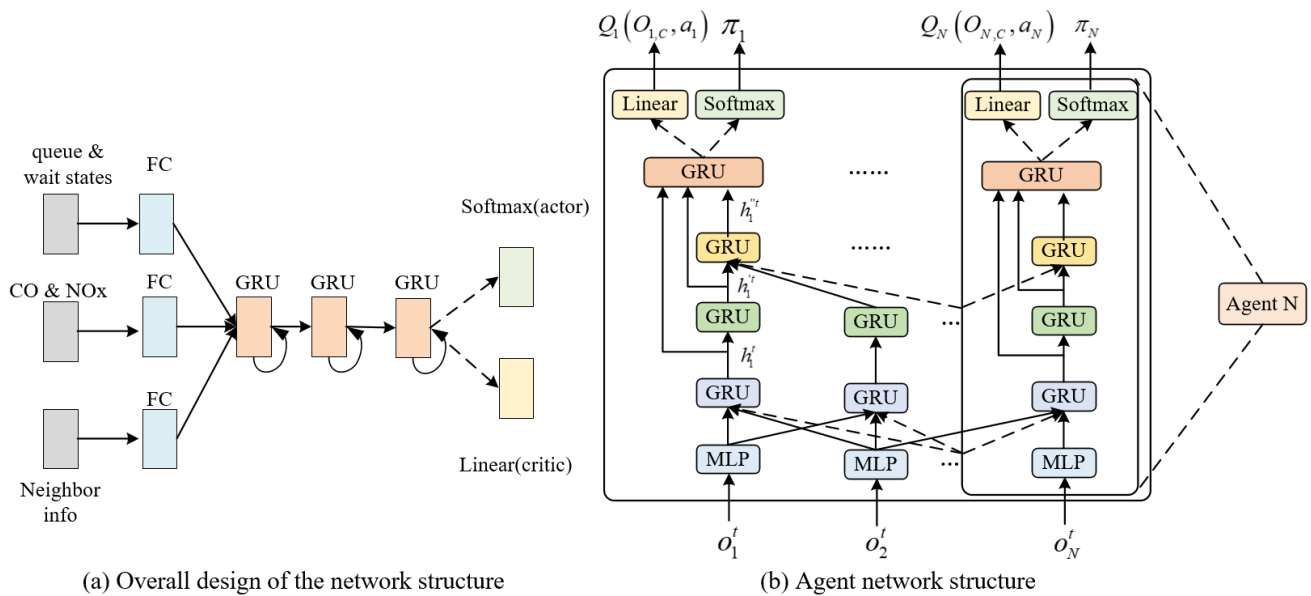


Figure 4. Agent network structure.

Figure 4b shows the design details of the agent network structure, where first, the local observation vector o_i^t acquired by the Agent is embedded into the MLP to obtain the intermediate value c_i^t , which is then fed into the GRU to obtain the hidden state h_i^t . Where $c_i^t = \sigma(o_i^t \cdot W_e + b_e)$, $h_i^t = GRU(c_i^t)$. W_e is the weight matrices, b_e is the bias vector, and the σ function is a Single Layer Perceptron (SLP) with the Relu's nonlinear activation function. Next, the second layer GRU extracts the traffic environment features of the merging intersection i and the area where its domain intersection is located to generate the implied feature vector h_i^{tT} . Continuing to stack GRUs, the perceptual domain of the implied feature vector becomes larger, and the traffic information embedded in the extracted

regional environmental features becomes more concentrated. That is, when stacking one GRU, the traffic state observation information acquired by the agent contains the feature vectors of the neighbor agents. When stacking two GRUs, the traffic state observation information acquired by agent i includes the feature vectors of the output of the first GRU and the state information acquired by the second GRU. However, no matter how many GRUs are superimposed, agent i only communicates with its neighbor agents, a feature that makes the method proposed in this paper better adapted to traffic signal control on urban road networks because each agent only plays a mixed-strategy Nash equilibrium game for a limited area rather than a strategy game for the entire urban road network.

4.2. Decision-Making Process Based on Mixed-Strategy Nash Equilibrium

Under distributed conditions, agents should consider not only the environment state but also the strategies of other agents when making decisions. The agent's strategy at game equilibrium is the optimal strategy to cope with other agents' strategies. When the MAS is in equilibrium, the distribution of agent's own experience is the mixed-strategy Nash equilibrium solution to cope with the neighboring agents. In order to obtain the mixed-strategy Nash equilibrium solution of the agent, which is the distribution of the agent's optimal decisions, this paper adopts Boltzmann's exploration to predict the strategy of neighboring agents and designs a decision-making process based on the mixed-strategy Nash equilibrium for the agent controlling the intersection. This decision-making mechanism not only accelerates the convergence of MARL but also improves the effect of traffic signal control in urban road networks and reduces the time delay and queue length of vehicles at intersections, as well as reducing the emission of pollutant gases and fuel consumption.

In order to make the MAS achieve the global optimal state when it reaches the final steady state, this paper constructs a general sum game based on the definition of agent's utility function of the game, the utility function of agent i is

$$u_i(a_i, a_{-i}, s_i) = \sum_{j \in I_{-i}} \omega_{l_{ji}}^1 \left(queue_{l_{ji}}^{pre}(a_i, a_{-i}, s_i) - queue_{l_{ji}} \right) + \omega_{l_{ji}}^2 \left(wait_{l_{ji}}^{pre}(a_i, a_{-i}, s_i) - wait_{l_{ji}} \right) \quad (2)$$

where $u_i(a_i, a_{-i}, s_i)$ denotes the game payoff function when agent i performs action a_i in the state s_i , and its neighbor agents perform the action a_{-i} , where agent j is the direct neighbor agent of the agent i , $j \in I_{-i}$; l_{ji} is the road section between the intersection controlled by the agent j and the intersection controlled by the agent i ; $queue_{l_{ji}}^{pre}(a_i, a_{-i}, s_i)$ is the expected vehicle queue length on the road section l_{ji} when agent i is in the state s_i and its own action a_i is responding to the action a_{-i} adopted by its neighbor agent j ; $queue_{l_{ji}}$ is the existing vehicle queue length on the road section l_{ji} ; $wait_{l_{ji}}^{pre}(a_i, a_{-i}, s_i)$ and $wait_{l_{ji}}$ are similar to $queue_{l_{ji}}^{pre}(a_i, a_{-i}, s_i)$ and $queue_{l_{ji}}$ in the previous section; $\omega_{l_{ji}}^1$ and $\omega_{l_{ji}}^2$ are both weighting factors indicating the extent to which the queue length and queue waiting time of section l_{ji} are in urgent need of improvement, respectively.

To obtain mixed-strategy Nash equilibrium solution, it is necessary to predict the strategy choice of neighborhood agents. When equilibrium is reached, the global of the MAS also realizes smooth convergence. At this time, the experience distribution of agent is the mixed-strategy Nash equilibrium solution for coping with neighboring agents. Therefore, in this paper, the MARL experience value approximation is used to replace the neighbor agents strategy choice expectation, which is to use Boltzmann exploration to predict the strategies of neighboring agents.

Assuming that agent i is in state s_i and its neighbor agent j is in state s_j , $a_j \in A_j(s_j)$ denotes the actions that can be taken by agent j in state s_j , and the basic components of the mixed strategy are denoted by the $Q_j(s_j, a_j)$ distribution:

$$Pr_j = \frac{\exp[Q_j(s_j, a_j)]}{\sum_{a_j \in A_j(s_j)} \exp[Q_j(s_j, a_j)]} \quad (3)$$

Let $\sigma_j \leftarrow Pr_j$ to obtain the mixed strategy σ_j ($\sigma_j \in \sigma_{-i}$) of neighbor agent j . The mixed-strategy Nash equilibrium solution of the agent i can be obtained using the known expected mixed strategies of neighborhood agents:

$$\sigma_i^* = \operatorname{argmax}_{\sigma_i} u_i(\sigma_i, \sigma_{-i}, s_i) \quad (4)$$

Then, the mixed-strategy Nash equilibrium solution σ_i^* of agent i can be obtained. Subsequently, the execution action a_i^{MSNE} of agent i is obtained by randomly sampling on the probability distribution $\hat{Pr}_i \leftarrow \sigma_i^*$.

4.3. Agent Learning Process Design

In order to realize the scalability of MARL traffic signal control and to address the non-smoothness of learning among agents, this paper adopts a learning framework with centralized training distributed execution to design the agent learning process, as shown in Figure 5. The agent learning specific process is divided into two parts. In the centralized training phase, the critic of each agent is trained based on the local observations and the observations and policies of neighboring agents. Then, the actor is trained to update with the help of Q-value and mixed-strategy Nash equilibrium solution given by the critic. In the testing phase, decisions are made using the trained actor decision function.

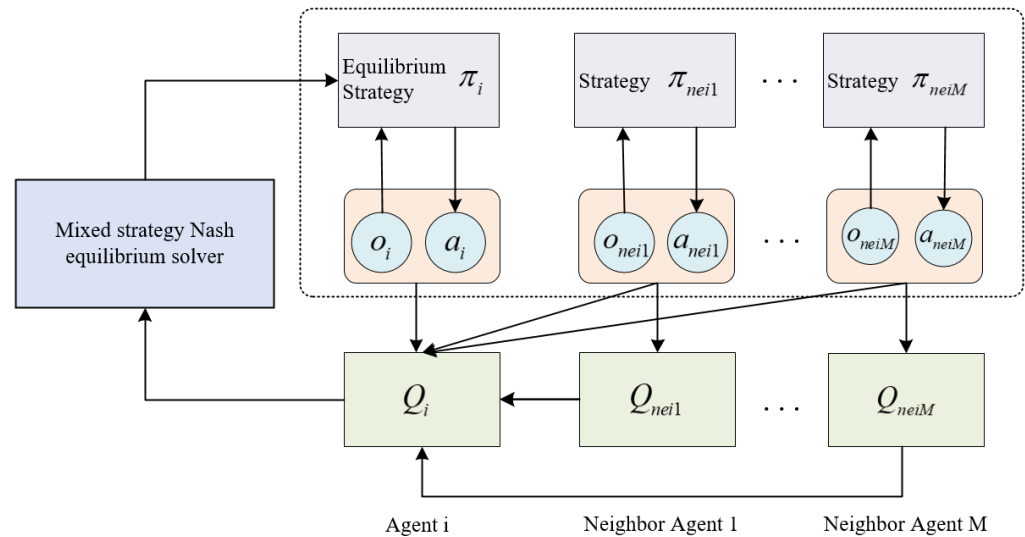


Figure 5. Agent learning process.

In order to break the correlation between the data and improve the experience utilization, the MADNPG algorithm adopts an experience replay mechanism. During training, the trained data tuples $(\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{O}')$ are first stored in the experience replay buffer D , where $\mathcal{O} = \{o_1, o_2, \dots, o_N\}$ is the set consisting of all agents' local observations, $\mathcal{A} = \{a_1, a_2, \dots, a_N\}$ is the set consisting of all agents' actions, $\mathcal{R} = \{r_1, r_2, \dots, r_N\}$ is the set consisting of all agents' rewards, and $\mathcal{O}' = \{o_1', o_2', \dots, o_N'\}$ is the set consisting of all agents' local observations at the next moment. Then, sample data of size B are randomly selected from D for end-to-end training from perception to action. In this case, the critic

network is trained by minimizing the squared value loss between the target Q value and the estimated Q value, and the value loss function of the agent i is given in Equation (5).

$$L(\theta_i) = \frac{1}{B} \sum_j \left[\left(Q_i^\pi(o_i^j, o_{nei1}^j, \dots, o_{neiM}^j; a_i^j, a_{nei1}^j, \dots, a_{neiM}^j) - y^j \right)^2 \right] \quad (5)$$

where $y^j = r_i^j + \gamma Q_i^{\pi'}(o_i^j, o_{nei1}^j, \dots, o_{neiM}^j; a_i^j, a_{nei1}^j, \dots, a_{neiM}^j) |_{a_k^j = \pi_k'(o_k^j)}$ is the target Q value, $Q_i^{\pi'}(o_i^j, o_{nei1}^j, \dots, o_{neiM}^j; a_i^j, a_{nei1}^j, \dots, a_{neiM}^j)$ denotes the Q value of agent i at the next moment, and γ is the reward discount factor.

Whereas, the actor network is trained by maximizing the cumulative expected return through gradient descent. The policy gradient of agent i can be written as

$$\nabla_{\theta_i} J(\pi_i) = \frac{1}{B} \sum_j \left[\nabla_{\theta_i} \pi_i(o_i^j) \cdot Q_i^\pi(o_i^j, o_{nei1}^j, \dots, o_{neiM}^j; a_i^j, a_{nei1}^j, \dots, a_{neiM}^j) |_{a_i^j = \pi_i(o_i^j)} \right] \quad (6)$$

where $\pi_i(o_i^j)$ denotes the action of agent i under observation o_i^j ; The evaluation function of agent i is $Q_i^\pi(o_i^j, o_{nei1}^j, \dots, o_{neiM}^j; a_i^j, a_{nei1}^j, \dots, a_{neiM}^j)$, its inputs are local observation information and neighbor's observation information, as well as local action and neighbor agents' action, and its output is the Q value of agent i . θ_i is the parameter of agent i 's strategy network.

4.4. MADNPG Algorithm

The goal of the DCNPG-TSC model is to train agents to learn an optimal joint equilibrium strategy in response to the real-time changing traffic flow in the road network. In order for the DCNPG-TSC model to produce the best equalization phase strategy for the next moment, this paper proposes the MADNPG algorithm, which finds the optimal network parameters for each agent by constantly updating the agent network parameters. In order to stabilize the learning process and improve the learning efficiency, the MADNPG algorithm also employs two techniques, the target network and the experience replay. Among them, the target network technique contains an evaluation network and a target network, both of which are obtained by the value function approximation network. The evaluation network first continuously interacts with the traffic environment and stores the interaction experience into the experience replay pool. Then, a batch of experiences is randomly drawn from the experience replay pool to train the evaluation network, and the parameters of each step are updated to the target network using soft update method. The target network updates the parameters of the evaluation network using gradient descent method according to the real label of the evaluation network.

The selection of the execution phase of each intersection is very important and needs to balance the utilization and exploration of RL. In the MADNPG algorithm, at the beginning of the training, it needs to spend more effort on exploration, and the exploration rate is set to $\varepsilon = 0.1$, and as the number of iterations increases, the exploration rate ε decreases to 0.01. The agent of the MADNPG algorithm chooses its local mixed-strategy Nash equilibrium action with probability $1 - \varepsilon$, which is

$$a_1, a_2, \dots, a_N = a_1^{MSNE}, a_2^{MSNE}, \dots, a_N^{MSNE} \quad (7)$$

Algorithm 1 gives the pseudo-code of the MADNPG algorithm. First, each local agent collects experiences according to the policy for reaching the mixed-strategy Nash equilibrium until enough experiences are collected to allow for small batch updates. Next, a small batch of experiences is randomly sampled to update each actor network and critic network. Then, backpropagation is performed using the first-order gradient optimizer Adam. Finally, the training process is terminated if the maximum step size is reached or a certain stopping condition is triggered.

Algorithm 1 Multi-Agent Distributional Nash Policy Gradient

Input: Agent set N , State Space S and Action Space A , Reward discount rate γ , Total simulation time T , batch data B , Actor network learning rate η_w , critic network learning rate η_θ

Output: trained parameters of target networks

- 1: Simulation time $t \leftarrow 0$, $B = \phi$
- 2: Initialize parameters ω, θ with random values
- 3: **for** Episode = 1 to M **do**
- 4: **while** $t < T_{max}$ **do**
- 5: **for** Each Agent $i \in 1, 2, \dots, N$ **do**
- 6: **for** Each Agent $i \in 1, 2, \dots, N$ **do**
- 7: Initialize sequence o_i^t
- 8: Receive Q_i^t
- 9: Find mixed-strategy Nash equilibrium solution σ_i^* based on the neighbor agent's Q_{nei}^t
- 10: Make decisions based on σ_i^* ($a_i^t \sim \sigma_i^*$) and random seed
- 11: **end for**
- 12: execute actions $a = (a_1^t, \dots, a_N^t)$ and observe new state o^t and receive the reward r^t
- 13: **if** $|\mathcal{D}| > \mathcal{D}_{max}$ **then**
- 14: Remove the oldest experiences in the memory
- 15: **end if**
- 16: Store transition $(\mathcal{O}, \mathcal{A}, \mathcal{O}', \mathcal{R})$ in $\mathcal{D}$
- 17: **if** $|\mathcal{D}| > B$ **then**
- 18: Select B samples (o^j, a^j, r^j, o'^j) from $\mathcal{D}$
- 19: $y^j = r_i^j + \gamma Q_i^{\pi'}(o_i^j, o_{nei1}^j, \dots, o_{neiM}^j; a_i^j, a_{nei1}^j, \dots, a_{neiM}^j) |_{a_k^j = \pi_k^j(o_k^j)}$
- 20: Update critic by minimizing the loss function using Equation (5)
- 21: Update actor using the sample policy gradient using Equation (6)
- 22: **end if**
- 23: **end for**
- 24: Update target network parameters for each Agent i : $\theta_i' = \tau\theta_i + (1 - \tau)\theta_i'$
- 25: **end while**
- 26: **end for**
- 27: **return** : $\theta_i, \theta_i' \ i \in 1, 2, \dots, N$

5. Simulation Experiments and Analysis of Results

SUMO [29], as an open source traffic simulation software, is not only powerful but also highly flexible. With SUMO, a variety of complex traffic simulation scenarios can be flexibly set up to meet specific simulation needs. Therefore, this paper adopts SUMO software (version: 1.3.1) to build a traffic simulation environment, and uses Tensorflow (version: 1.14.0) [30], a deep learning framework, to construct an agent network. To realistically and fairly compare the control effects of multiple traffic signal control methods with the DCNPG-TSC model proposed in this paper, two challenging experimental scenarios are designed which are (a) a synthetic traffic road network and (b) a real road network of 43 intersections in Jianye District, Nanjing City, Jiangsu Province, China, extracted from the real world using OpenStreetMap.

5.1. Baseline Methods

In order to verify the effectiveness and robustness of the DCNPG-TSC model in this paper, it is experimentally compared with the four most advanced RL-based traffic signal control methods. These methods include IQL-LR (Independent Q-Learning-Linear Regression), IQL-DNN (Independent Q-Learning-Deep Neural Network), IA2C (Independent Advantage Actor Critic) and MA2C (Multi-Agent Advantage Actor Critic). IQL is an independent RL method, where each agent learns its own strategy independently without

considering neighboring agents information and only makes decisions based on part of the traffic environment information it perceives. IQL-LR is a (linear regression, LR) LR-based IQL, and IQL-DNN is a DNN-based IQL, and the difference lies in the different methods of fitting the Q-function. IA2C extends the IQL idea on A2C (Advantage Actor–Critic) and employs a policy-based A2C to train agents, each of which independently interacts with the intersection environment. The MA2C method, on the other hand, builds on A2C by adjusting the neighbor fingerprints to improve the observability of the domain information, and by introducing a spatial discount factor to reduce the dimensionality of the traffic states and reward signals of the neighboring intersections, so that the agent can focus more on improving the traffic conditions in the region.

5.2. Experimental Parameter Setting

To ensure a fair comparison, all traffic signal control methods were performed in the same experimental environment and used the same action space, state space and interaction frequency. The experimental hyper-parameters include (1) SUMO hyper-parameters, such as, yellow light time t_y , simulation time per episode T_{max} ; (2) RL hyper-parameters, including the discount factor γ , experience replay buffer size D and batch random sampling size B ; and (3) optimization algorithm hyper-parameters, for instance, parameters α and β of the RMSProp optimization algorithm, which were set as experiential values. The specific settings of the parameters are shown in Table 2.

Table 2. Parameter settings in the DCNPG-TSC model.

Types	Hyper-Parameter	Description	Value
Simulator parameters	t_y	Yellow time interval	2
	T_{max}	Episode time length	3600
RL parameters	c_1	Weighting factor in Equation (1)	0.5
	c_2	Weighting factor in Equation (1)	0.5
	γ	Reward discount factor	0.99
	B	Batch size of training episode	20
	D	Size of replay memory	1×10^3
	$\omega_{l_{ji}}^1$	Weighting factor in Equation (1)	0.6
	$\omega_{l_{ji}}^2$	Weighting factor in Equation (1)	0.4
DL parameters	$\alpha_{RMSProp}$	RMSProp optimization parameter α	0.99
	$\beta_{RMSProp}$	RMSProp optimization parameter β	1×10^{-5}
	num_{MLP}	Number of MLP units in the encoding layer	64
	num_{Hidden}	Three hidden layer in the actor and critic Network	$64 + 64 + 64$
	num_{Output}	Output layer in the actor and critic Network	Number of Agents

5.3. 4×4 Synthesis Grid Net

The synthetic road network consists of 16 controlled intersections, as shown in Figure 6a, each of which is a six-entrance lane intersection with a two-lane east–west speed limit of 70 km/h and a single lane north–south speed limit of 40 km/h. The action space of each intersection contains four optional phases, which are S–N straight and right-turn phase, S–N left-turn phase, E–W straight and right-turn phase and E–W left-turn phase. In order to make the Markov Decision Process problem more challenging, the experiment simulates four time-varying traffic flow groups in the form of origin–destination, and at the beginning moment of the simulation, Router 1 is generated by the four traffic streams $x1$ – $x9$, $x2$ – $x10$, $x3$ – $x11$ and $x4$ – $x12$, and Router 2 is generated by the four traffic streams $x5$ – $x13$, $x6$ – $x14$, $x7$ – $x15$ and $x8$ – $x16$. In the figure, the blue solid line and the red dashed line represent the Router 1 and the Router 2, respectively, and the Router 3 and the Router 4

are their counterflow. Fifteen minutes after the beginning of the simulation, the traffic flow of Router 1 and Router 2 starts to decrease and the traffic flow of Router 3 and Router 4 start to be generated. Figure 6b shows the variation of the four traffic flows over time. The above defines the variation in the traffic flow with the simulation time, while the route of each vehicle is randomly generated at runtime without experimental setup.

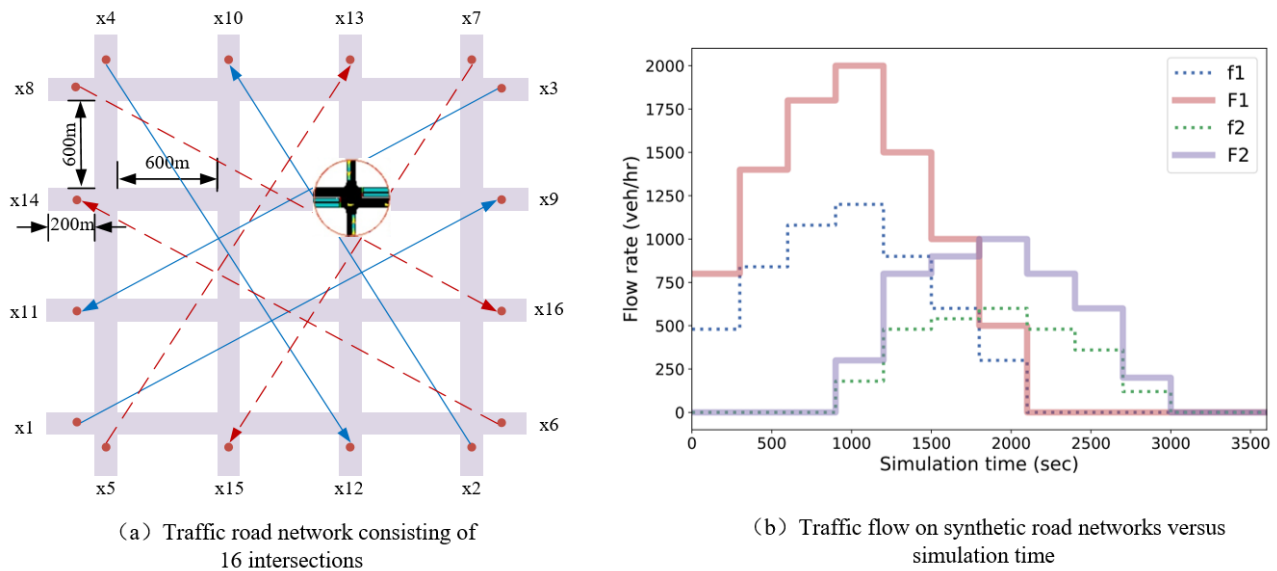


Figure 6. Experimental settings for synthetic road networks.

In order to reflect the overall congestion of the traffic road network, the experiment uses average queue length to evaluate the model. Figure 7 gives a graph of the average queue length in a synthetic road network with simulation time for five RL-based traffic signal control methods including the DCNPG-TSC model in this paper, where the solid line indicates the average value and the shading indicates the standard deviation of the evaluation set. As can be seen from Figure 7, the average queue lengths of IQL-DNN, IQL-LR, MA2C and IA2C show a gradual increasing trend, and the curve increases steeply after 900 s, which is due to the generation of convective traffic groups Router 3 and Router 4 leading to more and more traffic into the synthetic road network after 15 min. From the curves in Figure 7, it can be seen that the average queue length of the DCNPG-TSC methods is significantly smaller than the other four methods because the DCNPG-TSC method tends to choose a more stable mixed-strategy Nash equilibrium that exploits the information game at the neighborhood intersections to find a lower congestion level strategy.

Figure 8 plots the variation in average intersection delay with simulation time in the synthetic road network; the average delay curve is similar to the average queue length curve in that both show a gradual increase. From the figure, it can be seen that the delay curves of IQL-DNN and IQL-RL show a monotonically increasing trend and keep on increasing sharply with the simulation time, indicating that they are unable to recover from congestion. The delays in IA2C, MA2C and MADNPG increase slowly, and the MADNPG algorithm also outperforms the other MARL traffic signal control methods by consistently maintaining the lowest level of delay.

In order to further validate the effectiveness of each traffic signal control method, vehicle-based measurements are used in this paper to evaluate the effectiveness of various traffic signal control methods in terms of vehicle energy consumption and pollutant gas emissions. Table 3 summarizes the vehicle-related information measured for various RL-based traffic signal control methods in the synthetic road network, including the average vehicle speed (avg. speed, unit: m/s), average queue length (avg. queue, unit: m), average stopping time (avg. waits, unit: s), fuel consumption (unit: mL/s), and average CO and NOx emission (unit: mg/s). It can be seen that the IQL-DNN performs poorly in all

aspects due to the fact that IQL-DNN learns the traffic conditions independently while also suffering from high variance. The DCNPG-TSC model of the synthetic road network performs the best in all aspects, having the lowest average queue length, average stopping time, fuel consumption, and the least CO and NOx gas emissions, and the average speed of the vehicles is second only to the MA2C method. This is due to the fact that the DCNPG-TSC model learns a joint optimal policy to optimize signal control at intersections in urban road networks through a neighborhood game.

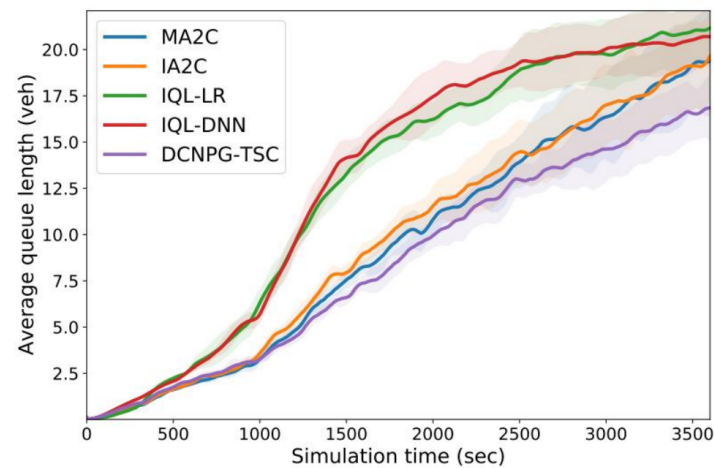


Figure 7. Variation in average queue length with simulation time in synthetic road networks.

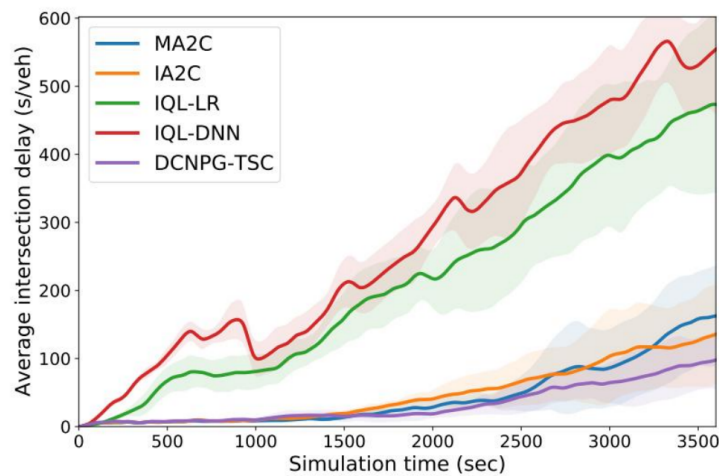


Figure 8. Average delay of intersections in synthetic road networks as a function of simulation time.

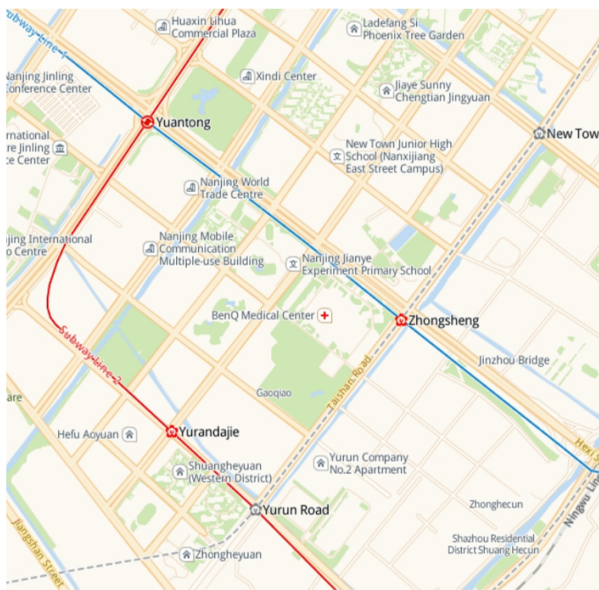
Table 3. Vehicle-based measurements in a 4×4 synthetic road network.

Metric	IQL-DNN	IQL-LR	IA2C	MA2C	DCNPG-TSC
Avg. Speed (m/s)	1.306	1.586	2.016	2.135	2.102
Avg. Queue (m)	13.138	12.874	9.765	9.447	8.580
Avg. Wait (s)	277.435	216.797	47.569	44.775	32.970
Fuel consumption (mL/s)	1.118	1.109	0.896	0.903	0.889
CO NOx emission (mg/s)	65.367	59.483	45.422	44.988	44.745

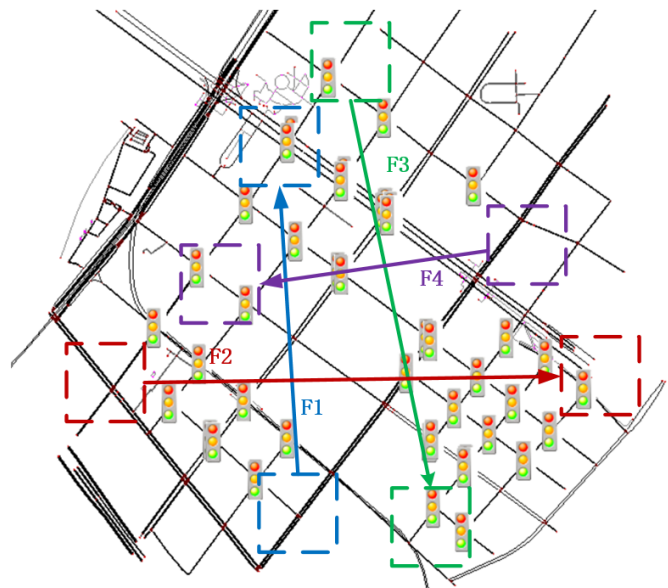
5.4. Real Urban Road Network

In order to verify the performance of the DCNPG-TSC model in a real road network, a real traffic road network in Jianye District, Nanjing City, Jiangsu Province, China, is selected for simulation experiments in this section, as shown in Figure 9a. This real road network has various types of roads and intersections, with a total of 43 controlled

intersections, 42 two-phase intersections and 1 three-phase intersection. In order to test the performance of the algorithm in the challenging Adaptive Traffic Signal Control (ATSC) scenario, dense, stochastic and time-varying traffic flows are designed to simulate rush hour traffic. Specifically, four traffic flow groups are generated from multiples of unit flow and randomly sampled origin–destination pairs in a given area, as shown in Figure 9b. Among them, F1 and F2 transform the simulated traffic flows every 5 min interval for the first 40 min, and the eight simulated flows transformed are [1, 2, 4, 4, 4, 4, 4, 2, 1] multiples of the unit flow, whereas the traffic flows F3 and F4 are generated starting from 15 min and ending the generation at 55 min.



(a) Real traffic area in Jianye District, Nanjing, Jiangsu Province, containing 43 controlled intersections. The road network map is derived from OpenStreetMap.



(b) Traffic road network in Jianye District, Nanjing, Jiangsu Province, with signalized intersections indicated by signals. The four traffic flow groups are indicated by arrows, with the start and end points located in the rectangular box.

Figure 9. Experimental settings for real urban road networks.

In order to verify the performance of the model in a complex real road network, simulation experiments on a real road network are conducted in this paper. Figure 10 plots the variation in average vehicle queue length with simulation time for various RL control methods in the real road network. From the figure, it can be seen that the IQL-LR and IQL-DNN methods perform poorly because their queue lengths show an incremental change with the simulation time, which cannot effectively inhibit the growth of queue lengths. The IA2C, MA2C, and DCNPG-TSC models all gradually reduce the vehicle queue lengths after reaching the inflection point, which indicates that these methods can effectively inhibit the queue lengths from increasing. Among them, the DCNPG-TSC model performs the best, presenting the lowest queue length.

In order to more accurately assess the performance of the DCNPG-TSC model, Figure 11 plots the average delay time of vehicles at intersections with the simulation time for various RL control methods in a real road network. From Figure 11, it can be seen that both IQL-DNN and IQL-LR methods perform poorly. MA2C, IA2C and DCNPG-TSC are all able to reduce delay time after peaks; however, both MA2C and IA2C fail to maintain consistently low intersection delays. This is because the upstream intersections of MA2C and IA2C greedily maximize their local flows leading to congestion, whereas DCNPG-TSC is more able to distribute traffic flows more evenly by gaming its neighboring intersections, which in turn leads to lower and more sustainable intersection delays.

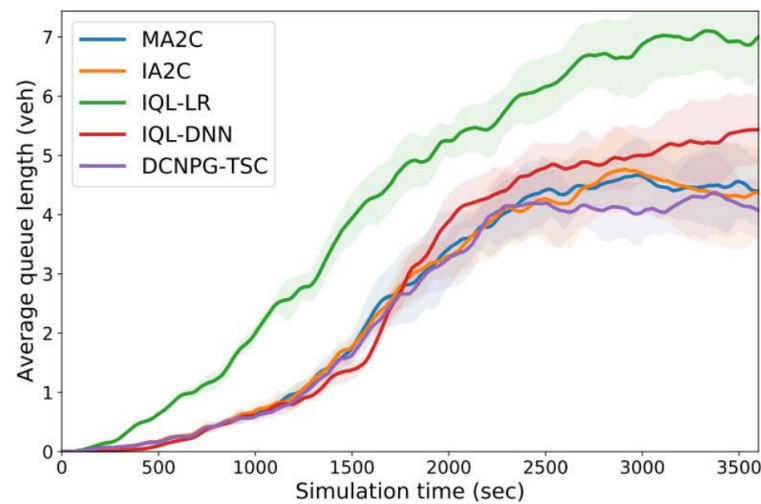


Figure 10. Average queue length at intersections in the real road network as a function of simulation time.

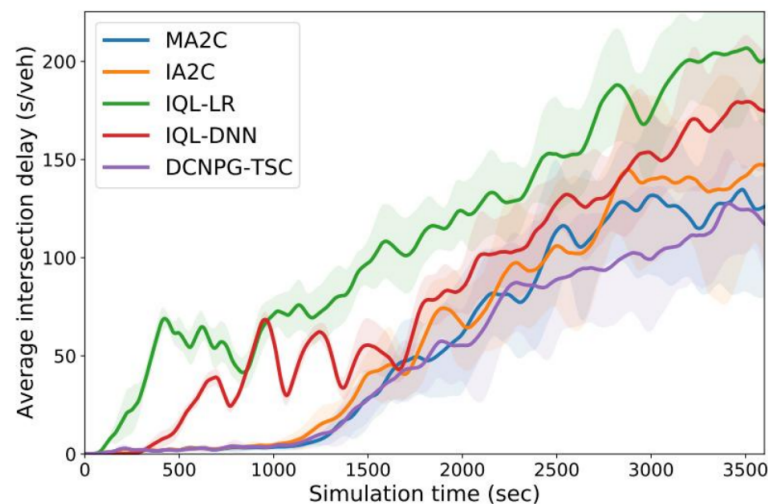


Figure 11. Variation of average intersection delay with simulation time in real road network.

In order to further validate the performance of the DCNPG-TSC model in this paper in terms of fuel consumption and pollutant gas emissions, experimental validation of each of the five methods is carried out, and the results are shown in Table 4. Table 4 summarizes the various traffic performance metrics of each RL method in the real road network. The fuel consumption of IA2C and MA2C are, respectively, reduced by 12.4% and 11.9% compared to the IQL-DNN method. The exhaust emissions of IA2C and MA2C are, respectively, reduced by 9.5% and 10.3% compared to the IQL-DNN, whereas DCNPG-TSC reduces the fuel consumption by 14.1% and exhaust emissions by 10% compared to IQL-DNN. It can be seen that the DCNPG-TSC model outperforms the other baseline methods in almost all the metrics, achieving the lowest average queue lengths and the lowest intersection waiting times, as well as the lowest fuel consumption and less CO and NO_x gas emissions. This is attributed to the fact that the agent network design allows the intersection agents to have access to a larger perceptual domain for better equilibrium gaming and collaboration, and the agent gaming mechanism results in a reasonable distribution of traffic flow, which optimizes the overall operation of the traffic system and improves traffic efficiency.

Table 4. Vehicle-based measurements in the real road network in Jianye District, Nanjing, Jiangsu Province, China.

Metric	IQL-DNN	IQL-LR	IA2C	MA2C	DCNPG-TSC
Avg. Speed (m/s)	4.877	2.414	3.827	3.904	3.946
Avg. Queue (m)	2.756	4.173	2.537	2.550	2.404
Avg. Wait (s)	84.413	114.932	64.075	58.634	52.646
Fuel consumption (mL/s)	1.446	1.299	1.266	1.274	1.242
CO NOx emission (mg/s)	59.070	67.462	53.477	52.944	53.179

6. Conclusions

To deal with the increasingly severe traffic congestion problem, this paper constructs a Deep Convolutional Nash Policy Gradient Traffic Signal Control (DCNPG-TSC) model, which is an adaptive multi-intersection traffic signal control model based on mixed-strategy Nash equilibrium and MARL. Specifically, firstly, through the design of the agent network structure, the DCNPG-TSC model can obtain more real-time traffic spatio-temporal state data and realize effective communication and collaboration of agents. Secondly, the Multi-Agent Distributional Nash Policy Gradient (MADNPG) algorithm used in the DCNPG-TSC model finds an optimal joint equalization strategy for each agent in the traffic road network to respond to the real-time changing traffic flow. Finally, the DCNPG-TSC model combines the green transportation concept with MARL traffic signal control and sets up a multi-optimization objective reward mechanism including reducing pollutant gas emissions and fuel consumption to reduce energy consumption and pollutant gas emissions. The experimental results of the synthetic road network and the real road network show that the DCNPG-TSC model proposed in this paper outperforms the other most advanced RL traffic signal control methods, which proves the effectiveness and scalability of the DCNPG-TSC model.

In spite of the progress in this paper, the research has the following limitations and constraints. Firstly, the gaming process takes more time, which affects the efficiency of model training; secondly, only the vehicle traffic at intersections is considered, and pedestrian traffic is not taken into account, which has some limitations in real traffic; thirdly, only the pollutant gases emitted by vehicles are focused on, and the particulate matter emitted by vehicles is ignored, which is an omission in comprehensively evaluating the pollutant emissions from vehicles.

Despite some limitations, this research still has important practical significance, especially for stakeholders such as traffic control centers and environmental agencies. For traffic control centers, the research can assist in designing more effective signal control strategies to reduce congestion and improve the efficiency of vehicular traffic. For environmental agencies, the research can help to accurately monitor the environmental impacts of vehicular pollutant gas emissions and develop effective environmental protection strategies to improve environmental quality and protect public health.

There is still some work required to be further studied. Firstly, the gaming process could be accelerated to reduce the gaming time consumed during the training of the DCNPG-TSC model and improve the training efficiency of the model. Secondly, more external data could be considered, such as pedestrians, weather, and so on, to further improve the performance of the DCNPG-TSC model. Thirdly, the measurement of vehicle pollutants, including particulate matter, can be improved to achieve a more accurate assessment of vehicle pollutant emissions.

Author Contributions: Conceptualization and supervision, methodology and writing—original draft, L.Y.; software, validation, formal analysis, visualization, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Natural Science Foundation of China (62362031, 62002117) and the Natural Science Foundation of Jiangxi Province (20224BAB202021).

Data Availability Statement: The datasets analyzed during the current study are available in the OpenStreetMap repository, <https://www.openstreetmap.org/> (accessed on 19 April 2023).

Acknowledgments: The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

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

Abbreviations

The following abbreviations are used in this manuscript:

DCNPG-TSC	Deep Convolutional Nash Policy Gradient Traffic Signal Control
MADNPG	Multi-Agent Distributional Nash Policy Gradient
SUMO	Simulation of Urban Mobility
MAS	Multi-Agent Systems
MARL	Multi-Agent Reinforcement Learning
RL	Reinforcement Learning
GRU	Gate Recurrent Unit

References

- Noaeen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Abad, Z.S.H.; Bazzan, A.L.; Far, B. Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Syst. Appl.* **2022**, *199*, 116830. [\[CrossRef\]](#)
- Yue, W.; Li, C.; Chen, Y.; Duan, P.; Mao, G. What is the root cause of congestion in urban traffic networks: Road infrastructure or signal control? *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8662–8679. [\[CrossRef\]](#)
- Shaikh, P.W.; El-Abd, M.; Khanafer, M.; Gao, K. A review on swarm intelligence and evolutionary algorithms for solving the traffic signal control problem. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 48–63. [\[CrossRef\]](#)
- Liu, P.; Qiao, Z.; Wu, Y.; Chen, K.; Hou, J.; Cai, L.; Tong, E.; Niu, W.; Liu, J. Traffic Signal Timing Optimization Based on Intersection Importance in Vehicle-Road Collaboration. In Proceedings of the International Conference on Machine Learning for Cyber Security, Singapore, 15–17 September 2023; pp. 74–89.
- Wang, W.; Chen, H.; Yin, G.; Mo, Y.; de Boer, N.; Lv, C. Motion State Estimation of Preceding Vehicles with Packet Loss and Unknown Model Parameters. *IEEE-ASME Trans. Mechatron.* **2024**, *23*, 48–63. [\[CrossRef\]](#)
- Aoki, S.; Rajkumar, R. Cyber traffic light: Safe cooperation for autonomous vehicles at dynamic intersections. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 22519–22534. [\[CrossRef\]](#)
- Lee, D.; He, N.; Kamalaruban, P.; Cevher, V. Optimization for reinforcement learning: From a single agent to cooperative agents. *IEEE Signal Process. Mag.* **2020**, *37*, 123–135. [\[CrossRef\]](#)
- Yan, L.; Zhu, L.; Song, K.; Zhu, L.; 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. [\[CrossRef\]](#)
- Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; Li, Z. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 3414–3421.
- Abdelghaffar, H.; Yang, H.; Rakha, H. Isolated traffic signal control using a game theoretic framework. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1496–1501.
- Liu, I.; Jain, U.; Yeh, R.; Schwing, A. Cooperative exploration for multi-agent deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 6826–6836.
- Abdelghaffar, H.M.; Rakha, H.A. Development and testing of a novel game theoretic de-centralized traffic signal controller. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 231–242. [\[CrossRef\]](#)
- Miao, C.; Chen, G.; Yan, C.; Wu, Y. An efficient algorithm for computing traffic equilibria using TRANSYT model. *Appl. Math. Modell.* **2010**, *34*, 3390–3399.
- Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; Li, Z. Colight: Learning network-level cooperation for traffic signal control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1913–1922.
- Wang, W.; 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. [\[CrossRef\]](#)
- Atta, A.; Abbas, S.; Abbas, M.A.; Ahmed, G.; Farooq, U. An adaptive approach: Smart traffic congestion control system. *Accid. Anal. Prev.* **2020**, *32*, 1012–1019. [\[CrossRef\]](#)
- Arel, I.; Liu, C.; Urbanik, T.; Kohls, A.C. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intel. Transp. Syst.* **2010**, *4*, 128–135. [\[CrossRef\]](#)

18. Li, S. Multi-agent deep deterministic policy gradient for traffic signal control on urban road network. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020; pp. 8243–8256.
19. 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. [\[CrossRef\]](#)
20. 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. [\[CrossRef\]](#)
21. Abdoos, M. A cooperative multiagent system for traffic signal control using game theory and reinforcement learning. *IEEE Intell. Transp. Syst. Mag.* **2020**, *13*, 6–16. [\[CrossRef\]](#)
22. Qu, Z.; Pan, Z.; Chen, Y.; Wang, X.; Li, H. A distributed control method for urban networks using multi-agent reinforcement learning based on regional mixed strategy Nash-equilibrium. *IEEE Access* **2020**, *8*, 19750–19766. [\[CrossRef\]](#)
23. Rasheed, F.; Yau, K.A.; Low, Y. Deep reinforcement learning for traffic signal control under disturbances: A case study on Sunway city, Malaysia. *Future Gener. Comput. Syst.* **2020**, *109*, 431–445. [\[CrossRef\]](#)
24. Lanctot, M.; Zambaldi, V.; Gruslys, A. A unified game-theoretic approach to multiagent reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4191–4204.
25. Wang, K.; Wang, Y.; Du, H.; Nam, K. Game-theory-inspired hierarchical distributed control strategy for cooperative intersection considering priority negotiation. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6438–6449. [\[CrossRef\]](#)
26. Wu, Q.; Wu, J.; Shen, J.; Du, B.; Telikani, A.; Fahmideh, M.; Liang, C. Distributed agent-based deep reinforcement learning for large scale traffic signal control. *Knowl.-Based Syst.* **2022**, *241*, 108304. [\[CrossRef\]](#)
27. Wei, W.; Wu, Q.; Wu, J.; Du, B.; Shen, J.; Li, T. Multi-agent deep reinforcement learning for traffic signal control with Nash Equilibrium. In Proceedings of the 2021 IEEE 23rd International Conference on High Performance Computing Communications, Haikou, China, 20–22 December 2021; pp. 1435–1442.
28. Li, C.H.; Shimamoto, S. A real time traffic light control scheme for reducing vehicles CO₂ emissions. In Proceedings of the 2011 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2011; pp. 855–859.
29. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wiefßner, E. Microscopic traffic simulation using sumo. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
30. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. TensorFlow: A system for Large-Scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

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.