sensors

MDPI

Article

Multi-Agent Decision-Making Modes in Uncertain Interactive Traffic Scenarios via Graph Convolution-Based Deep Reinforcement Learning

Xin Gao [1], Xueyuan Li [1,*], Qi Liu [1,*], Zirui Li [1,2], Fan Yang [1] and Tian Luan [1]

[1] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100080, China; 3120210298@bit.edu.cn (X.G.); 3120195255@bit.edu.cn (Z.L.); yangfanbitdb@163.com (F.Y.); 3220200285@bit.edu.cn (T.L.)

[2] Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands

* Correspondence: lixueyuan@bit.edu.cn (X.L.); 3120195257@bit.edu.cn (Q.L.)

Abstract: As one of the main elements of reinforcement learning, the design of the reward function is often not given enough attention when reinforcement learning is used in concrete applications, which leads to unsatisfactory performances. In this study, a reward function matrix is proposed for training various decision-making modes with emphasis on decision-making styles and further emphasis on incentives and punishments. Additionally, we model a traffic scene via graph model to better represent the interaction between vehicles, and adopt the graph convolutional network (GCN) to extract the features of the graph structure to help the connected autonomous vehicles perform decision-making directly. Furthermore, we combine GCN with deep Q-learning and multi-step double deep Q-learning to train four decision-making modes, which are named the graph convolutional deep Q-network (GQN) and the multi-step double graph convolutional deep Q-network (MDGQN). In the simulation, the superiority of the reward function matrix is proved by comparing it with the baseline, and evaluation metrics are proposed to verify the performance differences among decision-making modes. Results show that the trained decision-making modes can satisfy various driving requirements, including task completion rate, safety requirements, comfort level, and completion efficiency, by adjusting the weight values in the reward function matrix. Finally, the decision-making modes trained by MDGQN had better performance in an uncertain highway exit scene than those trained by GQN.

Keywords: multi-mode decision-making; connected autonomous vehicles; reward function matrix; uncertain highway exit scene; GQN; MDGQN

Citation: Gao, X.; Li, X.; Liu, Q.; Li, Z.; Yang, F.; Luan, T. Multi-Agent Decision-Making Modes in Uncertain Interactive Traffic Scenarios via Graph Convolution-Based Deep Reinforcement Learning. *Sensors* **2022**, *22*, 4586. https://doi.org/10.3390/s22124586

Academic Editors: Chao Huang, Yafei Wang, Peng Hang, Zhiqiang Zuo and Bo Leng

Received: 23 May 2022
Accepted: 15 June 2022
Published: 17 June 2022

1. Introduction

Artificial intelligence is in its golden development age due to the exponential increase in data production and the continuous improvements in computing power [1]. Autonomous driving is one of the main uses. A comprehensive autonomous driving system integrates sensing, decision-making, and motion-controlling modules [2–4]. As the "brains" of connected autonomous vehicles (CAVs) [5], the decision-making module formulates the most reasonable control strategy according to the state feature matrix transmitted by the sensing module, the vehicle state, and the cloud transmission information [6]. Moreover, it sends the determined control strategy to the motion-controlling module, including high-level behavior and low-level control requirements [7,8]. It is crucial to complete autonomous driving tasks safely and efficiently by making reasonable decisions based on other modules [9].

In an uncertain interactive traffic scenario, the driving environment has rigorous dynamic characteristics and high uncertainty, and the influences of driving behaviors of different traffic participants will be transmitted continuously [10]. On the level of transportation overall, all

traffic participants need to cooperate efficiently [11]. At the traffic participant level, individuals need to judge the risk factors and make appropriate decisions sensitively based on dynamic scene changes [12]. In [13], a Gaussian mixture model and important weighted least squares probability classifier were combined and used for scene modeling. That model can identify the braking strength levels of new drivers under the condition of insufficient driving data. The key to participating in traffic scenarios for CAVs is that each CAV needs to generate appropriate and cooperative behavior to match human vehicles and other CAVs. Therefore, CAVs urgently demand efficient and accurate multi-agent decision-making technology to effectively handle the interactions between different traffic participants.

The current multi-agent decision-making technologies mainly focus on deep reinforcement learning (DRL) due to the excellent performance of DRL in high-dimensional dynamic state space [14]. The keys of DRL in uncertain interactive traffic scenarios can be summarized as follows: (1) Efficient modeling of interactive traffic scenes and accurate representation of state features. (2) Generating reasonable and cooperative decision-making behaviors based on uncertain scene changes and individual task requirements. The design of the reward function is an essential part of the DRL application. Concretizing and numericizing task objectives realizes the communication between objectives and algorithms. Therefore, the design of the reward function determines whether the agent can generate reasonable and cooperative decision-making behaviors [15]. In addition, the accurate modeling of the interactive traffic environment and representation of state characteristics are the requirements for agents to generate expected behaviors.

In studies of traffic scenarios using the DRL method, researchers found that in uncertain interactive traffic scenarios, sparse reward problems lead to agents having a lack of effective information guidance [16], making the algorithm difficult to converge. In order to solve the sparse reward problem, researchers divide the original goal into different sub-goals and give reasonable rewards or punishments. In [17], the DDPG was adopted to settle the autonomous braking problem. The reward function was split into three parts to solve the problems: braking too early, braking too late, and braking too quickly. In [18], the reward function was divided into efficiency and safety rewards to train the comprehensive safety and efficiency decision model. In addition, considering the changes in driving task requirements and scene complexity, some studies have given different weights to the sub-reward functions based on the decomposition of the reward function, so as to train different decision-making modes. In [19], the reward function was divided into sub-reward functions based on security, efficiency, and comfort. The system can realize different control targets by adjusting the weight values in the reward function.

In the decision-making research involving uncertain interactive traffic scenarios, the DRL method only takes the individual characteristics of each vehicle as the input. It ignores the interactive influence of transitivity between vehicles. This will result in CAVs not generating reasonable and cooperative behavior, which may reduce total traffic efficiency and the occurrence of traffic accidents. Graph representation can accurately describe the interactions between agents, representing the relationship between vehicles in uncertain interactive traffic scenarios. Therefore, some researchers focused on the graph reinforcement learning (GRL) method and modeled the interaction with graph representation [20]. In [21], a hierarchical GNN framework was proposed and combined with LSTM to model the interactions of heterogeneous traffic participants (vehicles, pedestrians, and riders) to predict their trajectories. The GRL method combines GNN and DRL: the features of interactive scenes are processed by GNN, and cooperative behaviors are generated by the DRL framework [22]. In [23], the traffic network was modeled by dynamic traffic flow probability graphs, and a graph convolutional policy network was used in reinforcement learning.

This paper proposes an innovative, dynamic reward function matrix, and various decision-making modes can be trained by adjusting the weight coefficient of the reward function matrix. Additionally, the weight coefficient is further set as a function of reward, forming an internal dynamic reward function. In the traffic environment adopted in this paper, the randomness and interactions between HVs and CAVs are strengthened by using some human vehicles making uncertain lane changes. Two GRL algorithms are used in this

paper, GQN and MDGQN. Finally, we report a simulation based on the SUMO platform and a comparative analysis from various perspectives, such as reward functions, algorithms, and decision-making modes. The schematic diagram of the designed framework is shown in Figure 1.

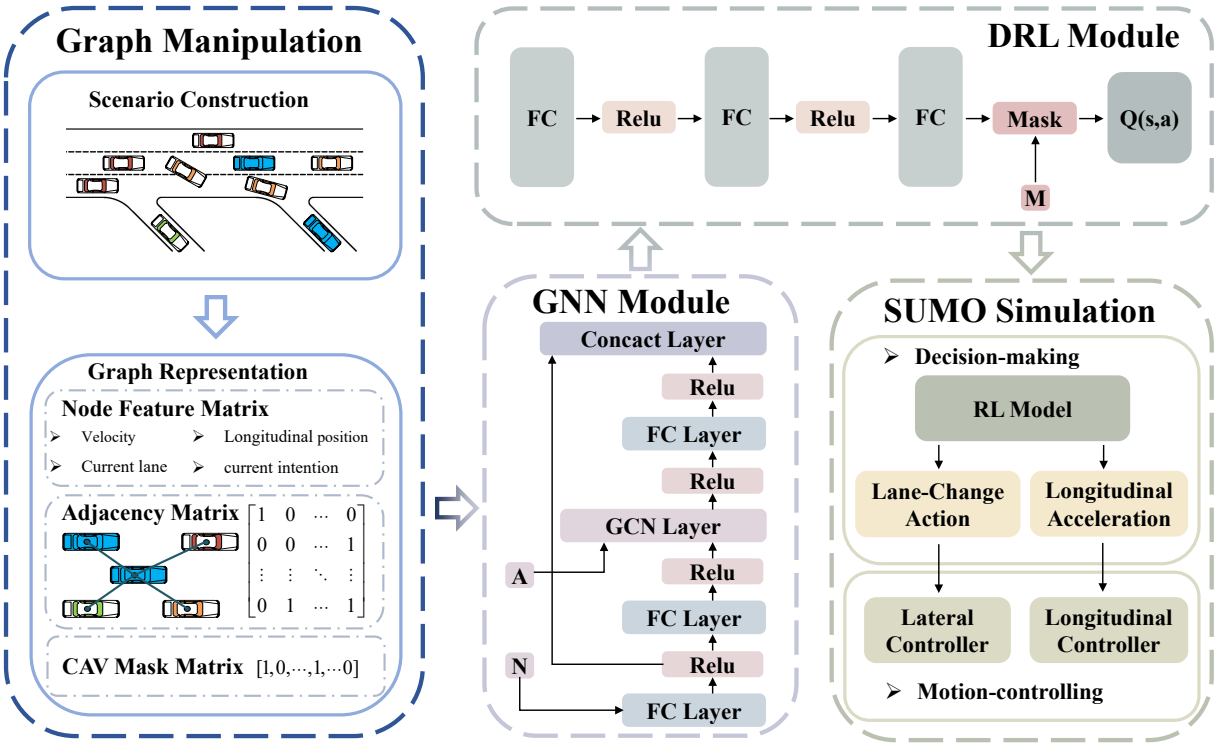

Figure 1. The schematic diagram of the designed framework. Letters N, A, and M represent node feature matrix, adjacency matrix, and CAV mask matrix respectively.

To summarize, the contributions of this paper are as follows:

1. Innovative dynamic reward function matrix: we propose a reward function matrix including a decision-weighted coefficient matrix, an incentive-punished-weighted coefficient matrix, and a reward–penalty function matrix. By adjusting the decision-weighted coefficient matrix, the decision-making modes of different emphases among driving task, traffic efficiency, ride comfort, and safety can be realized. Based on the premise that the incentive-punished function matrix separates the reward and the penalty, the optimization of individual performance can be achieved by adjusting the incentive and punishment ratio of each sub-reward function. In addition, the weight coefficient of the incentive-punished-weighted coefficient matrix is further set as a function of reward functions, which can reduce the impact of proportional adjustment on important operations, such as collision rate.
2. Adjust the parameters to train multiple decision-making modes: We compare the proposed reward function matrix with the traditional reward function under the same conditions. By adjusting the parameters of the decision-weighted coefficient matrix and the incentive-punished-weighted coefficient matrix, we can achieve aggressive or conservative incentive and punitive decision-making modes, respectively. Specifically, the four decision-making modes trained in this paper are the aggressive incentive (AGGI), aggressive punishment (AGGP), conservative incentive (CONI), and conservative punishment (CONP).
3. Modeling of interactive traffic scene and evaluation of decision-making modes: We designed a highway exit scene with solid interactions between CAVs and human vehicles (HVs) and adopted two algorithms to verify their differences. Addition-

ally, we also propose a set of indicators to evaluate the performance of driverless decision-making and used them to verify the performance differences among various algorithms and decision-making modes.

This article is organized as follows. Section 2 introduces the problem formulation. Section 3 introduces the methods used. Section 4 proposes the reward function matrix. Section 5 describes and analyzes the simulation results. Section 6 summarizes this paper and gives the future development directions.

2. Problem Formulation

2.1. Scenario Construction

As is shown in Figure 2, a 3-lane highway containing two exits was designed, and the three lanes were sorted from bottom to top as first, second, and third lanes. All the vehicles (HVs and CAVs) enter the road segment from the left of "Highway 0". The color of vehicle indicates its type and intention. In this paper, the vehicles were set as follows for safety principles and actual human driving rules:

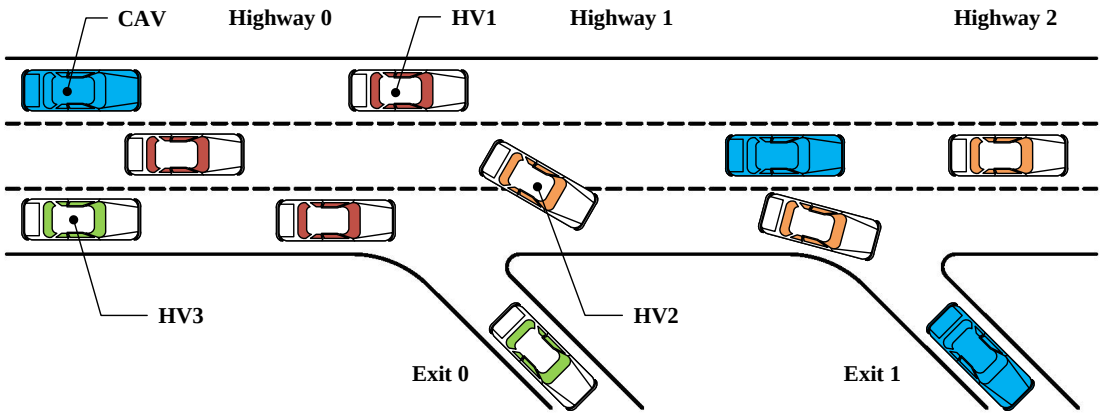

Figure 2. The highway exit scene with solid interactions between CAVs and HVs.

- A white body with a colored roof represents an HV, and an all-blue vehicle represents a CAV;
- The HV with the red roof (HV1) is set to appear in three lanes randomly and can only go out from "Highway 2";
- The HV with the orange roof (HV2) is set to emerge from the second lane and can go out from "Exit 1" when it is in the first lane and go out from "Highway 2" when it is in the other lanes;
- The HV with the green roof (HV3) is set to only appear from the first lane and can only go out from "Exit 0";
- When there is no car at the longitudinal distance of 10 m in the lane to be changed, the HV2 will be switched to the right;
- The CAV is set to appear in three lanes randomly and can go out from "Highway 2", "Exit 0", and "Exit 1".

The interaction between CAVs and HVs is enhanced through the above setting. The setting of HV1 enhances the uncertainty of the scene considerably, increasing the requirements for CAVs' decision-making. Straightening away from "Highway 2" is the most straightforward choice with the lowest collision risk, owing to CAVs' least lane change decisions. In addition, due to the uncertain lane change behavior of HV2, it is difficult for CAVs to explore "Exit 1" safely, and the collision risk of leaving the highway from "Exit 1" is higher than it is for other exits. Specific scenario setting parameters are shown in Table 1:

Table 1. Specific scenario setting parameters.

Parameter	Value	Parameter	Value
Length of "Highway 0"	150 m	Number of HV1	20
Length of "Highway 1"	200 m	Number of HV2	10
Length of "Highway 2"	150 m	Number of HV3	15
Time step	0.1 s	Number of CAVs	15
Initial velocity of HV1	24m/s	Probability of HV1 appearing every 0.1 s	0.18
Initial velocity of HV2	22 m/s	Probability of HV2 appearing every 0.1 s	0.08
Initial velocity of HV3	20 m/s	Probability of HV3 appearing every 0.1 s	0.12
nitial velocity of CAVs	20 m/s	Probability of CAVs appearing every 0.1 s	0.12

2.2. State Representation

The scenario is modeled as an undirected graph. Each vehicle in this scenario is regarded as the node of the graph, and the interaction between vehicles is considered the edge of the graph. Three matrices can represent the state space: a node features matrix X_t, an adjacency matrix A_t, and a CAV mask matrix M_t; each of them are described in the following.

Node Features Matrix: The vehicle's features are speed, position, location, and intention, denoted as $[V_i, Y_i, L_i, I_i]$. The node features matrix represents the features of each vehicle in the constructed scenario, which can be described as follows:

$$N_t = \begin{bmatrix} [V_1, Y_1, L_1, I_1] \\ [V_2, Y_2, L_2, I_2] \\ \cdots \\ [V_i, Y_i, L_i, I_i] \\ \cdots \\ [V_n, Y_n, L_n, I_n] \end{bmatrix} \tag{1}$$

where $V_i = v_{i-actual}/v_{\max}$ denotes the ratio of actual longitudinal velocity to maximum longitudinal velocity. $Y_i = y_{i-actual}/L_{highway}$ denotes the percentage of actual longitudinal position of the total length of the highway. L_i denotes the one-hot encoding matrix of the current lane of vehicles. I_i denotes the vehicle's one-hot encoding matrix of current intention (change to left lane, change to right lane, and go straight).

Adjacency matrix: In this paper, the interactions between vehicles are embodied in the information sharing between vehicles, expressed by the adjacency matrix. The calculation of the adjacency matrix is based on three assumptions:

- All CAVs can share information in the constructed scenario;
- Information cannot be shared between HVs;
- Vehicles can share information with themselves $a_{ii} = 1$.
- All CAVs can share information with HVs in their sensing range.

The derivation of the adjacency matrix is as follows:

$$A_t = \begin{bmatrix} a_{11} & a_{12} & \cdots & & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & & & \vdots \\ & & & a_{ij} & & \\ \vdots & \vdots & & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & & \cdots & a_{nn} \end{bmatrix} \tag{2}$$

where a_{ij} denotes the edge value of the ith vehicle and the jth vehicle. $a_{ij} = 1$ denotes that the ith vehicle and the jth vehicle share information; $a_{ij} = 0$ denotes that the ith vehicle and the jth vehicle share no information.

CAV mask matrix: CAV mask is used to filter out the embeddings of HVs after the GCN fusion block. The specific mathematical expression is as follows:

$$M_t = [m_1, m_2, \cdots, m_i, \cdots m_n] \tag{3}$$

where m_i = 0 or 1. If the ith vehicle is controlled by the GRL algorithm, $m_i = 1$; otherwise, $m_i = 0$.

2.3. Action Space

At each time step, each CAV has a discrete action space representing potential actions to be executed at the next time step, as follows:

$$a_i = \{a_{lane-change}, a_{acceleration}\} \tag{4}$$

where $a_{lane-change}$ indicates that the lane change action can be taken, including a left lane change, right lane change, or straight line; $a_{acceleration}$ denotes the discrete acceleration that CAV can take, and its value is equalized by interval $[-8\,\mathrm{m}\cdot\mathrm{s}^{-2}, 5\,\mathrm{m}\cdot\mathrm{s}^{-2}]$ at $0.5\,\mathrm{m}\cdot\mathrm{s}^{-2}$.

3. Methods

This section describes the principles of the methods used, including graph convolutional neural networks, Q-learning, GQN, and MDGQN.

3.1. Graph Convolutional Neural Network

GCN is a neural network model that directly encodes graph structure. The goal is to learn a function of features on a graph. A graph can be represented by $G = (V, E)$ in theory, where V denotes the set of nodes in the graph, the number of nodes in the graph is denoted by N, and E denotes the set of edges in the graph. The state of G is considered a tuple of 3 matrices of information: feature matrix X, adjacency matrix A, and degree matrix D.

- The adjacency matrix A is used to represent connections between nodes;
- The degree matrix D is a diagonal matrix, and $D_{ii} = \sum_j A_{ij}$;
- The feature matrix X is applied to represent node features, $X \in R^{N\times F}$, where F represents the dimensions of the feature.

GCN is a multi-layer graph convolution neural network, a first-order local approximation of spectral graph convolution. Each convolution layer only deals with first-order neighborhood information, and multi-order neighborhood information transmission can be realized by adding several convolution layers.

The propagation rules for each convolution layer are as follows [24]:

$$Z = g(H, A) = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)} + b) \tag{5}$$

where $\hat{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix; $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ and $W^{(l)}$ are layer-specific trainable weight matrices. $\sigma(\cdot)$ denotes an activation function, such as the $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. $H^{(l)} \in R^{N\times D}$ is the matrix of activations in the lth layer, $H(0) = X$.

3.2. Deep Q-Learning

Q-learning [25] is a value-based reinforcement learning algorithm. Q_t is the expectation that $Q(s_t, a_t)$ can obtain benefits by taking action $a_t (a_t \in A_t)$ under the state of $s_t (s_t \in S_t)$ at time t. The environment will feed back the corresponding reward R_t according to the agent's action. Each time step produces a quadruplet (s_t, a_t, r_t, s_{t+1}), and it is stored in the experience replay. Deep Q-learning [26] replaces the optimal action–value function with the deep neural network $Q(s, a, \omega)$. The following is a description of the principle of the algorithm.

The predicted values of DQN can be calculated according to the given four-tuple (s_t, a_t, r_t, s_{t+1}):

$$\hat{q}_t = Q(s_t, a_t, \omega) \tag{6}$$

The TD target can be calculated based on the actual observation reward r_t:

$$\hat{y}_t = r_t + \gamma \cdot \max_{a \in \mathcal{A}} Q(s_{t+1}, a, \omega) \tag{7}$$

DQN updates network parameters according to the following formula:

$$\omega \leftarrow \omega - \alpha \cdot (\hat{q}_t - \hat{y}_t) \cdot \nabla_\omega Q(s_t, a_t, \omega) \tag{8}$$

where α represents the learning rate.

3.3. Graph Convolutional Q-Network (GQN)

As described in Section 3.2, Q-learning uses a Q-Table to store the Q value of each state–action pair. However, if the state and action space are high-dimensionally continuous, there will be the curse of dimensionality; that is, with a linear increase in dimensions, the calculation load increases exponentially. GQN [27] replaces the optimal action–value function $Q_\star(s, a)$ with the graph convolutional neural network $Q(s, a, \theta)$:

$$Q(s, a, \theta) = \rho(Z, a) \tag{9}$$

where Z is the node embeddings output from graph convolution layer. ρ represents the neural network block, including the fully connected layer. θ is the aggregation of all the weights.

The specific training process of GQN is the same as DQN [26]. Firstly, the predictive value of GQN can be calculated according to the four-tuple (s_t, a_t, r_t, s_{t+1}) sampled from the experience replay:

$$\tilde{q}_t = Q(s_t, a_t, \theta_{now}) \tag{10}$$

However, since the Q-network uses the same estimate to update itself, it will cause bootstrapping and lead to deviation propagation. Therefore, another neural network can be used to calculate the TD target, called the target network $Q(s, a, \theta^-_{now})$. Its network structure is precisely the same as that of the Q-network, but the parameter θ^-_{now} is different from θ_{now}. Selecting the optimal action and forward propagation of the target network:

$$a^\star = \arg\max_{a \in \mathcal{A}} Q(s_{t+1}, a_t, \theta^-_{now}) \tag{11}$$

$$\tilde{q}_{t+1} = Q(s_{t+1}, a^\star, \theta^-_{now}) \tag{12}$$

Calculation of TD target $\hat{y}_t$ and TD error δ_t:

$$\hat{y}_t = r_t + \gamma \cdot \tilde{q}_{t+1} \tag{13}$$

$$\delta_t = \tilde{q}_t - \hat{y}_t \tag{14}$$

The gradient $\nabla_\theta Q(s_t, a_t, \theta_{now})$ is calculated by the backpropagation of a Q-Network, and the parameter of the Q-Network is updated by gradient descent:

$$\theta_{new} \leftarrow \theta_{now} - \alpha \cdot \delta_t \cdot \nabla_\theta Q(s_t, a_t, \theta_{now}) \tag{15}$$

where θ_{new} is the parameter of the updated Q-Network. α represents the learning rate.

Finally, GQN adopts the weighted average of the two networks to update the target network parameters:

$$\theta^-_{new} \leftarrow \tau \cdot \theta_{new} + (1 - \tau) \cdot \theta^-_{now} \tag{16}$$

where τ represents soft update rate.

3.4. Multi-Step Double Graph Convolutional Q-Network (MDGQN)

MDGQN further adopts double Q-learning and a multi-step TD target algorithm based on GQN. As shown in Section 3.3, the target network cannot completely avoid

bootstrapping, since the parameters of the target network are still related to the Q-Network. The double Q-learning algorithm is improved based on the target network. The Q-learning with the target network uses the target network to select the optimal action and calculate the Q value of the optimal action. However, the double Q-learning selects the optimal action according to the Q-Network and uses the target network to calculate the Q value of the optimal action. Equation (11) is modified as follows:

$$a^{\star} = \arg\max_{a \in \mathcal{A}} Q(s_{t+1}, a_t, \theta_{\text{now}}) \tag{17}$$

The multi-step TD target algorithm can balance the significant variance caused by Monte Carlo and the significant deviation caused by bootstrapping. Equation (13) is modified as follows:

$$\hat{y}_t = \sum_{i=0}^{m-1} \gamma^i r_{t+i} + \gamma^{\text{m}} \cdot \tilde{q}_{t+\text{m}} \tag{18}$$

where $\hat{y}_t$ is called the m-step TD target.

4. Reward Functions

The design of the reward function is an important criterion and goal of the DRL training process. Based on four aspects—the results the of the driving task, traffic efficiency, ride comfort, and safety—the reward function is divided into four blocks. Further, we propose a reward function matrix including a decision-weighted coefficient matrix, an incentive-punished-weighted coefficient matrix, and a reward–penalty function matrix.

$$\begin{aligned} r &= tr(A_{k1} A_{k2} A_r) \\ &= tr\left(\begin{bmatrix} \begin{bmatrix} k_r & & & \\ & k_e & & \\ & & k_c & \\ & & & k_s \end{bmatrix} \begin{bmatrix} k_{rI} & & & & & & & \\ k_{rP} & & & & & & & \\ & k_{eI} & & & & & & \\ & k_{eP} & & & & & & \\ & & & k_{cI} & & & & \\ & & & k_{cP} & & & & \\ & & & & & k_{sI} & & \\ & & & & & k_{sP} & & \end{bmatrix}^T \begin{bmatrix} r_{result-I} & & & & & & & \\ r_{result-P} & & & & & & & \\ & r_{efficiency-I} & & & & & & \\ & r_{efficiency-P} & & & & & & \\ & & & r_{comfort-I} & & & & \\ & & & r_{comfort-P} & & & & \\ & & & & & & r_{safe-I} & \\ & & & & & & r_{safe-P} & \end{bmatrix} \end{bmatrix} \end{aligned} \tag{19}$$

Specific parameters are described below:

1. A_{k1} is the decision-weighted coefficient matrix. k_r, k_e, k_c, and k_s denote the weights of reward and penalty functions based on the results of the driving task, traffic efficiency, ride comfort, and safety.
2. A_{k2} is the incentive-punished-weighted coefficient matrix. k_{rI}, k_{eI}, k_{cI}, and k_{sI} denote the weights of reward functions based on the results of the driving task, traffic efficiency, ride comfort, and safety, respectively. k_{rP}, k_{eP}, k_{cP}, and k_{sP} denote the weights of penalty functions based on the results of the driving task, traffic efficiency, ride comfort, and safety, respectively.
3. A_r is a reward–penalty function matrix. $r_{result-I}$, $r_{efficiency-I}$, $r_{comfort-I}$, and r_{safe-I} are reward functions based on the results of the driving task, traffic efficiency, ride comfort, and safety, respectively. $r_{result-P}$, $r_{efficiency-P}$, $r_{comfort-P}$, and r_{safe-P} are penalty functions based on the results of the driving task, traffic efficiency, ride comfort, and safety, respectively.

By adjusting the weight coefficient of the decision-weighted coefficient matrix and incentive-punished-weighted coefficient matrix, DRL can train different goal-oriented decision-making modes. In the decision-making process of autonomous vehicles, the upper control module can choose different decision-making modes according to different needs. In order to select a decision-making model with excellent comprehensive performance and strong contrast, we conducted multiple sets of experiments, and some experimental data were put in Appendix A. This paper determined four decision modes: AGGI, AGGP, CONI, and CONP, by adjusting the parameters in Tables 2 and 3.

Table 2. Weight coefficient of the decision-weighted coefficient matrix.

Parameters	Conservative	Aggressive
The weight of task k_r	0.2	0.25
The weight of efficiency k_e	0.2	0.35
The weight of comfort k_c	0.25	0.15
The weight of safety k_s	0.35	0.25

Table 3. Weight coefficient of the incentive-punished-weighted coefficient matrix.

Parameters	Incentive	Punished
The weight of incentive sub-reward function k_{rI0} , k_{eI0} , k_{cI0} , k_{sI0}	0.6	0.4
The weight of punished sub-reward function.k_{rP0} , k_{eP0} , k_{cP0} , k_{sP0}	0.4	0.6

Since the change of weight coefficient will dilute some essential rewards or punishments, this paper improves the reward function against this defect. The weight coefficient of the incentive-punished-weighted coefficient matrix is further set as a functional of reward functions, which forms an internal dynamic reward function. The specific formula is as follows:

$$\begin{cases} k_{rI} = k_{rI0} \cdot \exp\Big((r_{result-P} + r_{comfort-P})/100{,}000\Big) \\ k_{eI} = k_{eI0} \cdot \exp\Big((r_{result-P} + r_{comfort-P})/80{,}000\Big) \\ k_{sI} = k_{sI0} \cdot \exp\Big((r_{result-P} + r_{comfort-P})/150{,}000\Big) \end{cases} \tag{20}$$

Based on [3], the specific reward functions and penalty functions were designed. Firstly, we designed the corresponding reward function and penalty function based on the results of the driving tasks. The independent variable of the reward function is the number of CAVs and HV2 reaching destinations, which aims to train decisions that can assist HVs in completing driving tasks. The penalty function is designed based on collisions.

$$r_{result-I} = 300(n_{r1} + n_{r2}) \tag{21}$$

$$r_{result-P} = -60{,}000, if\ collision \tag{22}$$

where n_{r1} is the number of CAVs leaving from "Exit 1". n_{r2} is the number of CAVs leaving from "Exit 1".

To train the decision-making model to improve traffic efficiency, this paper divides the speed interval of CAVs into three parts. The corresponding reward and penalty functions were designed to curb speeding, encourage high-speed driving, and punish low-speed blocking for these three-speed ranges. In order to make CAVs faster and more stable to explore the optimal speed, we used the exponential function to design a soft reward function [28].

$$r_{efficiency-I} = \exp\left[\frac{6 \times (v_y - v_{y\min})}{v_{y\max}}\right], if\ v_{y\min} \le v_y \le v_{y\max} \tag{23}$$

$$r_{efficiency-P} = \begin{cases} -\exp(v_y - v_{y\max}), if\ \ v_y > v_{y\max} \\ -\exp\Big(1 + \frac{v_{y\min} - v_y}{3}\Big), if\ v_y < v_{y\min} \end{cases} \tag{24}$$

where v_y is the velocity of the CAV. $v_{y\max}$ represents the maximum velocity allowed by the current lane; its value is 25 m $\cdot$ s^{-1}, or 15 m $\cdot$ s^{-1}.

In order to improve the ride comfort of all vehicles in this traffic section, the corresponding reward function and penalty function are designed based on the acceleration and lane change times of all vehicles.

$$r_{comfort-I} = \exp(3) \times n_{c1} \tag{25}$$

$$r_{comfort-P} = -2000 \times n_{c2} - \exp\left(\frac{m}{2}\right) \tag{26}$$

where n_{c1} is the number of vehicles with acceleration of $[-2\ \mathrm{m}\cdot\mathrm{s}^{-2}, 2\ \mathrm{m}\cdot\mathrm{s}^{-2}]$. n_{c2} is the number of vehicles with acceleration of $(-\infty, -4.5\ \mathrm{m}\cdot\mathrm{s}^{-2}]$. m is the number of lane changes in this traffic section within 0.5 s.

Superior security performance is the premise of developing decision technology [29]. In [30], the length of CAVs' safety time is one of the most important factors affecting road safety. This paper introduces the safety time of the CAVs into the corresponding reward function. The definition of the safety time is as follows:

$$t_1 = t_{safe-follower} = \frac{y_{AV} - y_{follower}}{v_{follower} - v_{AV}} \tag{27}$$

$$t_2 = t_{safe-leader} = \frac{y_{leader} - y_{AV}}{v_{AV} - v_{leader}} \tag{28}$$

where y_{AV} is the longitudinal position of the CAV. y_{leader} and $y_{follower}$ are the longitudinal positions of the front and rear vehicles of CAV, respectively. v_{leader} and $v_{follower}$ are the longitudinal speeds of the front and rear vehicles of CAV, respectively.

A driving hazard diagram is proposed to represent the degree of danger of the vehicle's state based on safety time $t_{safe-follower}$ and $t_{safe-leader}$. As shown in Figure 3, three primary colors are used to represent the degrees of danger in this state. The red region represents collision accident danger, and the deeper the color, the greater the likelihood. The yellow area indicates that the vehicle needs to pay attention to the occurrence of a possible emergency. The green area indicates that the vehicle is in a safe state. By dividing Figure 3 into five categories, the sub-reward functions were designed. On the basis of the above principles, a security-based reward function is proposed for training security decisions. The formula is shown in Table 4.

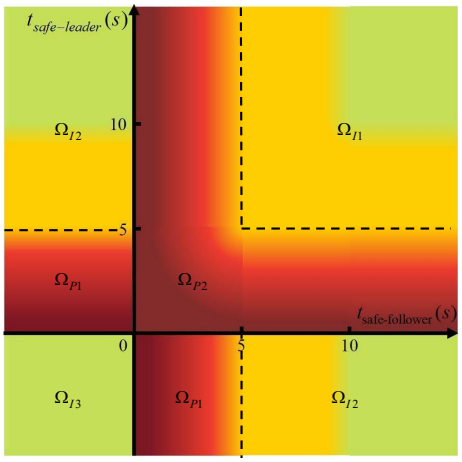

Figure 3. The driving hazard diagram.

Table 4. The security-based reward function.

Subfunction	Category	Calculation
r_{safe-I}	Ω_{I1}	$\max\left(\exp(\frac{\min(t_1,t_2)-5}{2.5}), \exp(2)\right)$
	Ω_{I2}	$\max\left(\exp(\frac{\max(t_1,t_2)-5}{2.5}), \exp(2)\right)$
	Ω_{I3}	$\exp(2)$
r_{safe-P}	Ω_{P1}	$-\exp(4-\min(t_1,t_2))$
	Ω_{P2}	$-\exp(4-\max(t_1,t_2))$

5. Experiments

5.1. Parameter Setting

In this section, we show the solid interaction scene designed through SUMO. Firstly, based on this scene, the proposed reward function is compared with the traditional reward function. The two algorithms were used to train four decision-making modes, and the differences between different algorithms and modes are compared. Finally, the performances of four decision-making modes based on the two algorithms are evaluated. The main parameters for algorithms are listed in Table 5.

Table 5. Parameters of the graph reinforcement learning.

Parameters	Value
Optimizer	Adam
Nonlinearity	Relu
Learning rate	0.005
Discount factor	0.99
Updating rate	0.05
Minibatch size	32
Multi-step	3

5.2. Evaluation Indexes

In order to further evaluate the test performance of each decision-making mode, four kinds of evaluation indexes are proposed, namely, efficiency index, safety index, comfort index, and task index.

1. Efficiency: The average longitudinal velocity of CAVs and all vehicles in this scenario is proposed and used to evaluate the traffic efficiency of CAVs and the comprehensive efficiency of total traffic flow under different decision modes. Their functions are defined as follows:

$$\bar{v}_{AVs} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} v_{AV-ij}}{N} \tag{29}$$

$$\bar{v}s. = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}}{N_{all}} \tag{30}$$

   where v_{AV-ij} represents the longitudinal velocity of the ith CAV detected at the jth time step. v_{ij} is the longitudinal velocity of the ith vehicle detected at the jth time step. N indicates the total number of CAVs' longitudinal velocities detected in all-time steps. N_{all} indicates the total number of all vehicles' longitudinal velocities detected in all-time steps.
2. Safety: Due to the presence of multiple CAVs in the test scenario, we define the collision rate as the probability of each CAV collision accident in each episode.

$$p_{collision} = \frac{N_{collision}}{N_{CAV}} \tag{31}$$

   where $N_{collision}$ is the number of collisions in a single episode. N_{CAV} represents the number of CAVs.
3. Comfort: For the evaluation of comfort, we mainly studied the deceleration of all vehicles under different decision-making modes. All vehicles' total emergency braking times $N_{braking}$ in a single episode are defined as comfort evaluation indexes. It should be noted that the deceleration between $(-\infty, -4.5\ \mathrm{m}\cdot\mathrm{s}^{-2}]$ is regarded as emergency braking.

4. Task: Based on the solid interaction between CAVs and HVs in the test scenario, the driving task completion rates of CAVs and HV2 are used as the task indexes. Their functions are defined as follows.

$$p_{CAV} = \frac{N_{CAV-Exit1}}{N_{CAV}} \tag{32}$$

$$p_{HV2} = \frac{N_{HV2-Exit1}}{N_{HV2}} \tag{33}$$

where $N_{CAV-Exit1}$ is the number of CAVs that left via "Exit 1". $N_{HV2-Exit1}$ indicates the number of HV2 that left via "Exit 1". N_{HV2} represents the number of HV2.

5.3. Validation of the Reward Function

The proposed reward function was used in GQN and MDGQN to train different decision-making modes while improving training efficiency. In [27], the traditional reward function only considers intention reward, speed reward, lane-changing penalty, and collision penalty. In this article, this traditional reward function is used as the baseline, and the reward of the CONP decision-making mode training process is compared. In order to allow them to make a fair comparison based on the maximum and minimum reward values they can achieve, the reward values in the training process are normalized.

As shown in Figure 4, using the proposed reward function for training can explore the maximum reward rapidly. The traditional reward function's training process experiences repeated reward reductions, and the maximum reward cannot be explored in the specified number of rounds. In addition, the training stability is greatly improved by comparing the reward fluctuation in the training process with the baseline. After the 200th episode, the average normalized reward values of Baseline-GCQ, CONP-GCQ, and CONP-MDGCQ were 0.7399, 0.9882, and 0.9886, respectively; and their variances were 0.0871, 0.0028, and 0.0049. Under the condition of using the GQN algorithm, the CONP decision-making model was 35.40% better than the average of baseline, and the variance is only 3.27% higher than that of the baseline. In summary, the proposed reward function can promote the fast convergence of the algorithm and greatly enhance the stability of the training process.

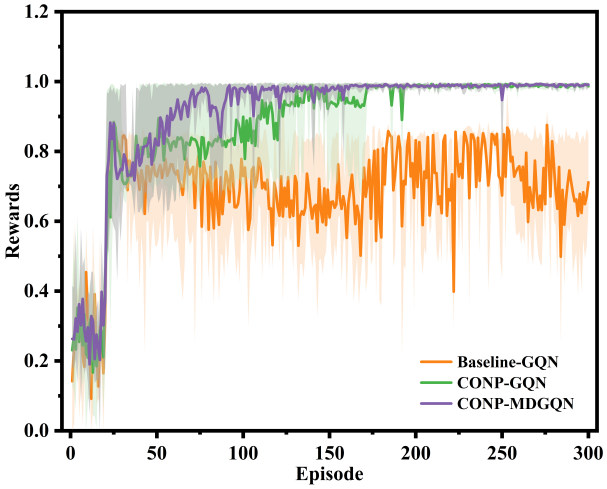

Figure 4. The comparison between the proposed reward function and baseline.

5.4. Training of Different Decision-Making Modes

As shown in Figure 5, under the four decision-making modes, the MDGQN algorithm converged faster than GQN, and the fluctuation of reward decline decreased significantly. This verified that MDGQN effectively alleviates the overestimation problem after using the multi-step TD target and double Q-learning algorithm to explore the optimal action more quickly. Based on MDGQN algorithm analysis, the AGGI decision-making mode converged in the 80th episode. In contrast, the AGGP decision-making mode did not fully

converge until the 160th episode, which further verifies that the reward and punishment ratio of the reward function affects the convergence speed of the algorithm. In summary, by comparing the four decision-making modes, the aggressive decision-making mode uses more punishment than the conservative decision-making mode. The incentive decision-making mode can promote the convergence of the algorithm faster than the punished decision-making mode.

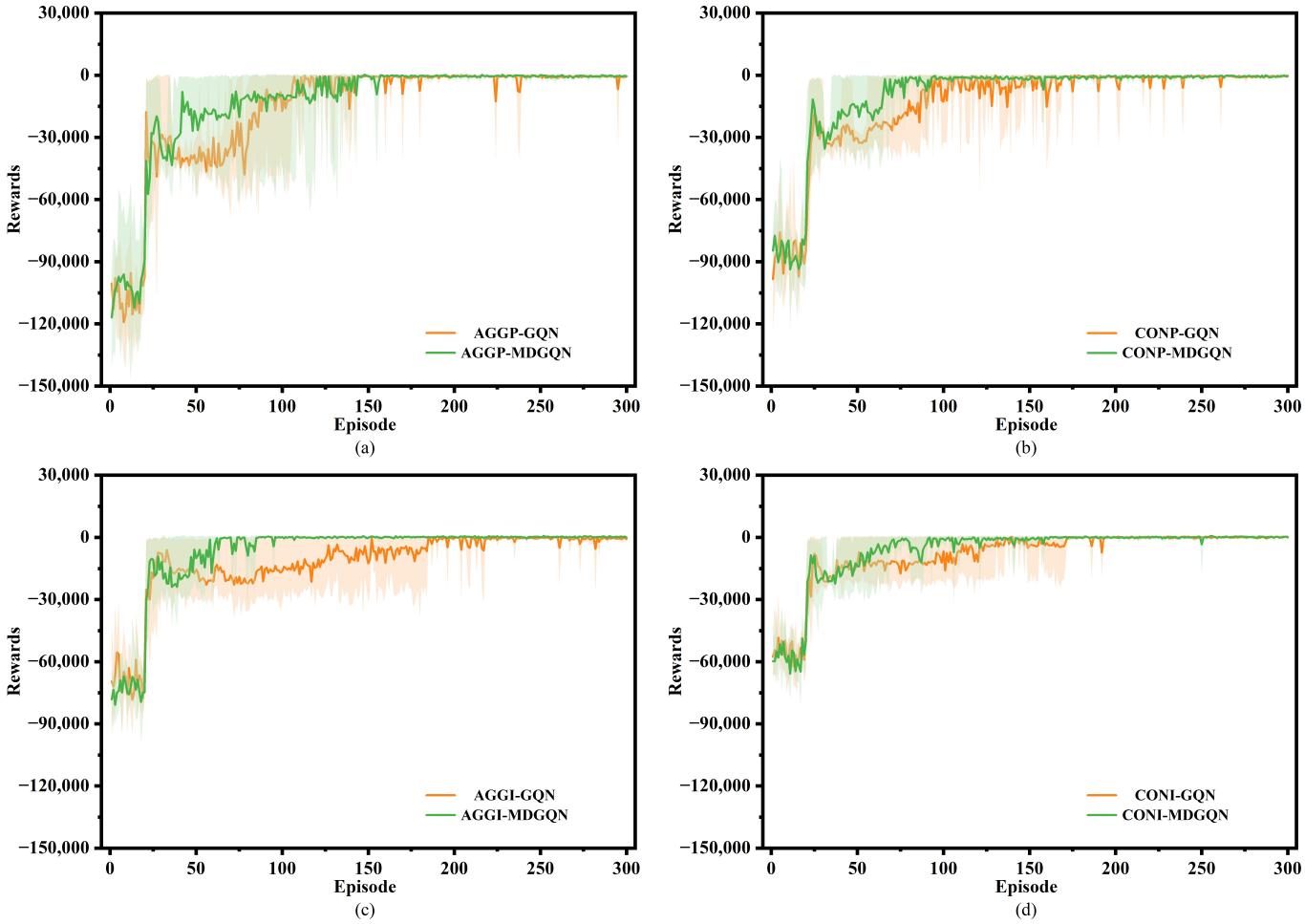

Figure 5. The training reward diagrams of four modes are compared and verified by the GQN and MDGQN algorithms. (**a**) The AGGP decision-making mode, (**b**) the CONP decision-making mode, (**c**) the AGGI decision-making mode, and (**d**) the CONI decision-making mode.

Furthermore, the training decision-making mode was tested, and the reward mean and variance are compared. As shown in Figure 6, MDGQN explored higher reward thresholds and averages than GQN in all four decision modes. Under AGGP, AGGI, CONP, and CONI decision-making modes, the corresponding test reward variances of MDGQN were 93.91, 80.26, 64.71, and 87.10; these were less than 108.51, 108.19, 93.78, and 97.01 for GQN. It can be concluded that the test stability of the CONP decision mode is the strongest, but the ability to distinguish algorithm differences is poor. AGGI is the decision-making mode that can best reflect the differences between algorithms, but the test stability decreases slightly.

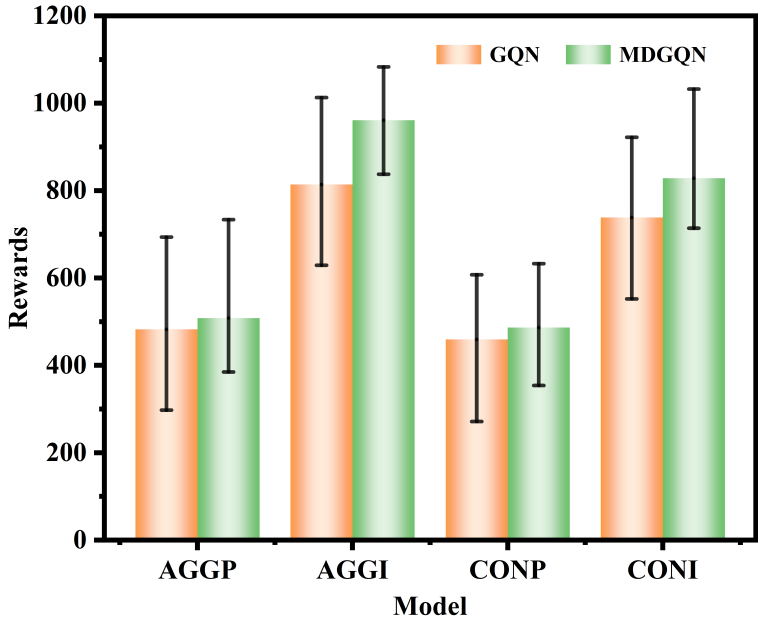

Figure 6. Test reward diagram of four decision-making modes. Orange represents the decision-making mode using the GQN algorithm, and green indicates the decision-making mode using the MDGQN algorithm. In the figure, the two ends of the error rod represent the maximum and minimum values of the test values.

5.5. Evaluation of Decision-Making Modes

Based on Figure 7, the two algorithms are first compared. In the four modes, the average speed of CAVs was better than that of GQN except for the CONI decision mode. MDGQN performed better than GQN in the speed of total traffic flow under the four modes. However, the proportion of CAVs' average speed increase in the total traffic flow's average speed increase is less than 1. In the CONI decision-making mode, the average speed of CAVs trained by MDGQN was lower than that of GQN, but the total traffic flow speed increased. This shows that based on the designed reward function, the MDGQN algorithm can obtain decisions that improve the total traffic efficiency by training CAVs. As shown in Table 6, MDGQN had a lower collision rate, fewer emergency braking times, and a higher arrival rate of CAVs and HV2 than GQN under the same decision-making mode, except for a slight increase in the number of emergency braking incidents under CONP decision-making mode. This can be analyzed from the corresponding data. Under the CONP decision-making mode, the p_{HV2} of MDGQN was 0.717, which was 18.12% higher than that of GQN. When the HV2 increases in "Exit 1", the uncertainty of the scene will enhance, increasing the number of emergency brake events. From the above results, it can be concluded that the comprehensive performance of the MDGQN algorithm is superior to that of GQN in this specific autonomous driving scene, and the MDGQN is more suitable for solid, interactive, uncertain scenes.

Additionally, the four decision-making modes are compared. Compared with the conservative decision-making mode, the aggressive decision-making mode had a higher average speed, a higher collision rate, more emergency braking incidents, and a higher vehicle arrival rate. Compared with the punished decision-making model, the average speed of the incentive decision-making mode was slightly higher, the collision rate was higher, the number of emergency braking incidents was higher, and the arrival rate of HV2 was further improved. Among the four decision-making modes, AGGI decision-making mode had the highest traffic efficiency; CAVs and HV2 had the highest arrival rate, but their collision rate and number of emergency braking incidents were are also the highest. The collision rate and emergency braking incidents of CONP decision mode were the lowest, but the traffic efficiency was the lowest, and the arrival rates of CAVs and HV2 were the lowest. Those performance differences are consistent with the weight coefficient

allocation of the proposed reward function matrix. This fully proves that the proposed reward function matrix can effectively train various decision-making modes to adapt to autonomous driving scenarios.

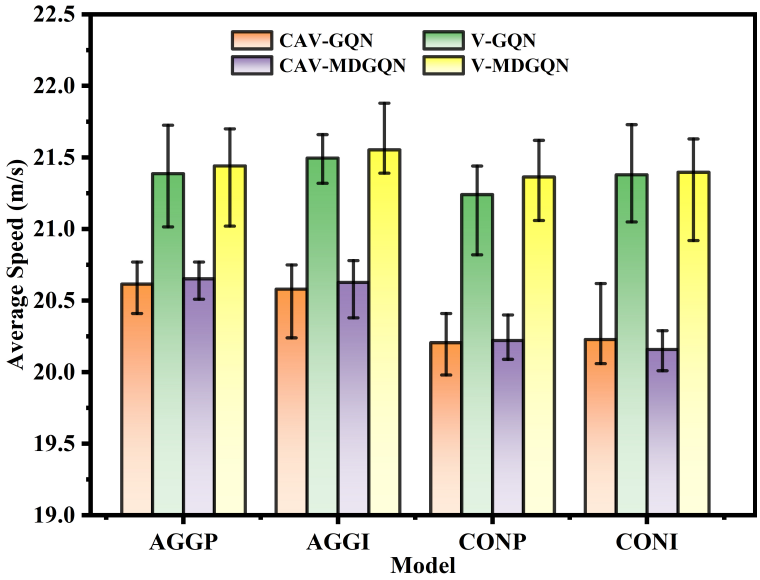

Figure 7. Average longitudinal velocity of CAVs and total traffic flow under four decision-making modes. The CAV-GQN means the average longitudinal speed of CAVs under the training of the GQN algorithm. The CAV-MDGQN represents the average longitudinal velocity of CAVs under the MDGQN algorithm training. The V-GQN indicates the average longitudinal speed of the total traffic flow under the training of the GQN algorithm. The V-MDGQN indicates the average longitudinal velocity of total traffic flow under MDGQN training. The two ends of the error rod indicate the maximum and minimum values of the test values.

Table 6. Test evaluation results.

Modes	Algorithm	$p_{collision}$	$N_{braking}$	p_{CAV}	p_{HV2}
AGGP	GQN	0.0111	3.1	0.991	0.763
	MDGQN	0.0067	1.867	1	0.813
AGGI	GQN	0.0133	3.567	1	0.93
	MDGQN	0.0022	2.533	1	0.937
CONP	GQN	0.0045	1.667	0.987	0.607
	MDGQN	0.0	1.9	0.996	0.717
CONI	GQN	0.0111	2.5	1	0.857
	MDGQN	0.0045	2.3	1	0.83

6. Conclusions

We proposed a reward function matrix, including a decision-weighted coefficient matrix, an incentive-punished-weighted coefficient matrix, and a reward–penalty function matrix. By adjusting the weight coefficient of the reward function matrix, various decision-making modes can be trained. We trained four decision-making modes, namely, AGGI, AGGP, CONI, and CONP; and the GQN algorithm and the MDGQN algorithm based on GQN improvement were used for verification by comparison. A large number of simulation results proved the following three conclusions. Firstly, the proposed reward function can promote the fast convergence of the algorithm and greatly improve the stability of the training process. Taking the CONP decision-making mode as an example, the average normalized reward value after the 200th round was 35.40% higher than that of the baseline, and the variance was only 3.27% greater than that of the baseline. Secondly,

the comprehensive performance of the MDGQN algorithm is superior to that of GQN. Under the four decision-making modes, the averages and variances of test reward values of MDGQN are better than those of GQN. In terms of driving performance, MDGQN performs better than GQN, except that the number of brakes increases slightly in CONP decision-making mode. Finally, the proposed reward function matrix can effectively train various decision-making modes to adapt to different autonomous driving scenarios. With an increase in incentive weight, the comparison effect of the algorithm is more obvious, but the security will decrease. In our future work, we will further study the interactions of autonomous vehicles and decision mode switching.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

In the training process of decision-making mode, we adjusted different weight coefficients to find a mode with high comprehensive performance. To better validate the effectiveness of the proposed reward function matrix at training different decision patterns, we list four other sets of parameter settings, as shown in the Table A1.

Table A1. Parameter settings.

Modes	k_r	k_e	k_c	k_s	k_I0	k_P0
(a)	0.1	0.1	0.2	0.6	0.6	0.4
(b)	0.4	0.4	0.05	0.15	0.6	0.4
(c)	0.25	0.35	0.15	0.25	0.75	0.25
(d)	0.2	0.2	0.25	0.35	0.7	0.3

The simulation results are shown in Table A2. From this we can conclude that too much of an increase in the proportion of safety will lead to a decline in the completion rate of the driving tasks, especially CAVs' loss of the driving task objectives. Excessive increases in the proportions of the driving tasks and efficiency can lead to a rapid decline in safety. An increase in incentive rewards will lead to a decrease in security, and higher task completion rate and efficiency. Based on the above results, four groups of reasonable parameters were selected, and four decision models, CONP, CONI, AGGP, and AGGI, were trained.

Table A2. Test evaluation results.

Modes	$\bar{v}$	$p_{collision}$	$N_{braking}$	p_{CAV}	p_{HV2}
(a)	20.872	0.033	1.967	3.733	10.967
(b)	21.747	1.367	7.833	9.733	15
(c)	21.581	0.333	4.367	9.633	15
(d)	21.41	0.233	2.8	8.933	15

References

1. Stoma, M.; Dudziak, A.; Caban, J.; Droździel, P. The Future of Autonomous Vehicles in the Opinion of Automotive Market Users. *Energies* **2021**, *14*, 4777. [CrossRef]
2. Liu, Q.; Li, X.; Yuan, S.; Li, Z. Decision-Making Technology for Autonomous Vehicles Learning-Based Methods, Applications and Future Outlook. In Proceedings of the IEEE International Intelligent Transportation Systems Conference, Indianapolis, IN, USA, 19–22 September 2021.
3. Chen, L.; He, Y.; Wang, Q.; Pan, W.; Ming, Z. Joint optimization of sensing, decision-making and motion-controlling for autonomous vehicles: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4642–4654. [CrossRef]
4. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
5. Liu, Q.; Yuan, S.; Li, Z. A Survey on Sensor Technologies for Unmanned Ground Vehicles. In Proceedings of the 2020 3rd International Conference on Unmanned Systems, Harbin, China, 27–28 November 2020. [CrossRef]
6. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Souza, A. Self-driving cars: A survey. *Expert Syst. Appl.* **2020**, *165*, 113816. [CrossRef]
7. Yu, Y.; Lu, C.; Yang, L.; Li, Z.; Gong, J. Hierarchical Reinforcement Learning Combined with Motion Primitives for Automated Overtaking. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020.
8. Kłosowski, G.; Gola, A.; Amila, T. Computational Intelligence in Control of AGV Multimodal Systems. *IFAC-PapersOnLine* **2018**, *51*, 1421–1427. [CrossRef]
9. Liu, Q.; Li, Z.; Yuan, S.; Zhu, Y.; Li, X. Review on Vehicle Detection Technology for Unmanned Ground Vehicles. *Sensors* **2021**, *21*, 1354. [CrossRef]
10. Bouton, M.; Nakhaei, A.; Fujimura, K.; Kochenderfer, M.J. Cooperation-aware reinforcement learning for merging in dense traffic. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3441–3447.
11. Caban, J.; Nieoczym, A.; Dudziak, A.; Krajka, T.; Stopková, M. The Planning Process of Transport Tasks for Autonomous Vans–Case Study. *Appl. Sci.* **2022**, *12*, 2993. [CrossRef]
12. Nieoczym, A.; Caban, J.; Dudziak, A.; Stoma, M. Autonomous vans - the planning process of transport tasks. *Open Eng.* **2020**, *10*, 18–25. [CrossRef]
13. Li, Z.; Gong, J.; Lu, C.; Li, J. Personalized Driver Braking Behavior Modelling in the Car-following Scenario: An Importance Weight-based Transfer Learning Approach. *IEEE Trans. Ind. Electron.* **2022**, *69*, 10704–10714. [CrossRef]
14. Schwarting, W.; Alonso-Mora, J.; Rus, D. Planning and Decision-Making for Autonomous Vehicles. *Annu. Rev. Control Robot. Auton. Syst.* **2018**, *1*, 187–210. [CrossRef]
15. Matignon, L.; Laurent, G.J.; Fort-Piat, N.L. *Reward Function and Initial Values: Better Choices for Accelerated Goal-Directed Reinforcement Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
16. Ou, X.; Chang, Q.; Chakraborty, N. Simulation study on reward function of reinforcement learning in gantry work cell scheduling. *J. Manuf. Syst.* **2019**, *50*, 1–8. [CrossRef]
17. Fu, Y.; Li, C.; Yu, F.R.; Luan, T.H.; Zhang, Y. A Decision-Making Strategy for Vehicle Autonomous Braking in Emergency via Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5876–5888. [CrossRef]
18. Wang, H.; Gao, H.; Yuan, S.; Zhang, F.; Li, K. Interpretable Decision-Making for Autonomous Vehicles at Highway On-Ramps with Latent Space Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8707–8719. [CrossRef]
19. Chen, Q.; Zhao, W.; Li, L.; Wang, C.; Chen, F. ES-DQN: A Learning Method for Vehicle Intelligent Speed Control Strategy under Uncertain Cut-in Scenario. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2472–2484. [CrossRef]
20. Peng, Y.; Tan, G.; Si, H.; Li, J. DRL-GAT-SA: Deep reinforcement learning for autonomous driving planning based on graph attention networks and simplex architecture. *J. Syst. Archit.* **2022**, *126*, 102505. [CrossRef]
21. Li, Z.; Lu, C.; Yi, Y.; Gong, J. A Hierarchical Framework for Interactive Behaviour Prediction of Heterogeneous Traffic Participants Based on Graph Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–13. [CrossRef]
22. Jiang, J.; Dun, C.; Huang, T.; Lu, Z. Graph Convolutional Reinforcement Learning. *arXiv* **2018**, arXiv:1810.09202.
23. Peng, H.; Du, B.; Liu, M.; Liu, M.; He, L. Dynamic Graph Convolutional Network for Long-Term Traffic Flow Prediction with Reinforcement Learning. *Inf. Sci.* **2021**, *578*, 401–416. [CrossRef]
24. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
25. Watkins, C.J.C.H. Learning from Delayed Rewards. Ph.D. Thesis, Kings College University of Cambridge, Cambridge, UK, 1989.
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjel, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
27. Dong, J.; Chen, S.; Ha, P.; Li, Y.; Labi, S. A DRL-based Multiagent Cooperative Control Framework for CAV Networks: A Graphic Convolution Q Network. *arXiv* **2020**, arXiv:2010.05437.
28. Li, G.; Yang, Y.; Li, S.; Qu, X.; Lyu, N.; Li, S.E. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. *Transp. Res. Part Emerg. Technol.* **2022**, *134*, 103452. [CrossRef]

29. Khayyam, H.; Javadi, B.; Jalili, M.; Jazar, R.N. Artificial Intelligence and Internet of Things for Autonomous Vehicles. In *Nonlinear Approaches in Engineering Applications: Automotive Applications of Engineering Problems*; Jazar, R.N., Dai, L., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 39–68. [CrossRef]
30. Tarkowski, S.; Rybicka, I. Distraction of the Driver and Its Impact on Road Safety. *Transp. Res. Procedia* **2020**, *44*, 196–203. [CrossRef]