World Electric Vehicle Journal

MDPI

Article

Vehicle Safety Planning Control Method Based on Variable Gauss Safety Field

Zixuan Zhu [1], Chenglong Teng [2], Yingfeng Cai [3,*], Long Chen [1], Yubo Lian [2] and Hai Wang [1]

[1] Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China
[2] BYD Auto Industry Co., Ltd., Shenzhen 518118, China
[3] School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China
* Correspondence: caicaixiao0304@126.com

Abstract: The existing intelligent vehicle trajectory-planning methods have limitations in terms of efficiency and safety. To overcome these limitations, this paper proposes an automatic driving trajectory-planning method based on a variable Gaussian safety field. Firstly, the time series bird's-eye view is used as the input state quantity of the network, which improves the effectiveness of the trajectory planning policy network in extracting the features of the surrounding traffic environment. Then, the policy gradient algorithm is used to generate the planned trajectory of the autonomous vehicle, which improves the planning efficiency. The variable Gaussian safety field is used as the reward function of the trajectory planning part and the evaluation index of the control part, which improves the safety of the reinforcement learning vehicle tracking algorithm. The proposed algorithm is verified using the simulator. The obtained results show that the proposed algorithm has excellent trajectory planning ability in the highway scene and can achieve high safety and high precision tracking control.

Keywords: autonomous driving; planning algorithm; variable Gaussian safety field; reinforcement learning; policy gradient

Citation: Zhu, Z.; Teng, C.; Cai, Y.; Chen, L.; Lian, Y.; Wang, H. Vehicle Safety Planning Control Method Based on Variable Gauss Safety Field. *World Electr. Veh. J.* **2022**, *13*, 203. https://doi.org/10.3390/wevj13110203

Academic Editors: Yong Li, Xing Xu, Lin Zhang, Yechen Qin and Yang Lu

Received: 12 October 2022
Accepted: 26 October 2022
Published: 31 October 2022

1. Introduction

In recent years, autonomous driving technology has developed rapidly due to its significant economic potential and advantages in improving traffic efficiency and driving safety. Various methods have been proposed to solve the decision-making problem of autonomous vehicles in highway driving tasks. Most studies have considered decision making as a control problem. As an unavoidable part of the autonomous driving system, trajectory planning is of great significance to the study of the autonomous vehicle. Avoiding the surrounding obstacles accurately and driving safely and efficiently based on the upper perception and prediction results are the basic requirements for automobile driving. Therefore, most autonomous driving researchers are now focusing on more intelligent, safe and efficient trajectory-planning methods.

The existing trajectory-planning methods are generally divided into four categories: potential field methods [1], sample-based methods [2], search-based methods [3], and optimization-based methods [4]. A potential field method simulates the movement of a controlled object in space into a forced movement of a particle in a virtual force field and plans the future trajectory of a vehicle by calculating the combined force field to which the vehicle is subjected. However, this method relies on accurate modeling of the environment, which will put the training into the dilemma of the local optimal solution and increase the computational cost. The sampling-based methods are mainly divided into fast random search tree (RRT) and probability path map (PRM) methods. The probability map path method is based on the graph structure, converts the continuous space into a discrete space, and uses the search algorithms such as A* to find paths on the route map

to improve search efficiency. However, this method needs to solve the boundary value problem and does not focus on generating paths in the process of building the graph. The search-based planning algorithms mainly refer to map search methods, including A*, D*, and the corresponding variants. This kind of algorithm is widely used in the field of robot motion planning, but its planned path does not consider the geometric constraints of the road and has poor smoothness. Qi Xuanxuan et al. [5] introduced simulated annealing to optimize the expansion of nodes and heuristic functions, and guided the algorithm to search for the target point, which improved the inefficiency of the traditional A* algorithm but still fell into the dilemma of a suboptimal solution. To improve sampling efficiency and avoid suboptimal dilemmas for agents, Claussmann et al. [6] classified the spatial configuration for route planning into three main categories: sampling [7], connection unit [8], and raster representation (Lattice) [9]. The raster representation can be used to predict and plan based on the moving obstacles around the vehicle while considering the kinematic constraints. However, the raster method is difficult to sample completely and can only sample better driving tracks. It is also difficult for the complete search method to consider the dynamic constraints of the automobile. The trajectory planning based on the optimization method has higher computational power requirements for the vehicle computer, and the optimization delay between each frame is large. In summary, most of the existing traditional trajectory-planning methods have relatively stable security performance and excellent computational efficiency. However, they focus only on the generation of the optimal path and can fall into the suboptimal dilemma.

In recent years, deep reinforcement learning (DRL) has shown satisfactory performance in both trajectory planning and trajectory tracking control. Feher et al. [10] trained deep deterministic policy gradient (DDPG) agents to generate waypoints for vehicle tracking and achieved good results. However, the algorithm only focused on the lateral trajectory and provided a suboptimal solution. Several studies have used original sensor measurements to generate turn angles and throttle values [11–16] in an end-to-end manner. The deep deterministic actor-critic (DDAC) algorithm [11,12] can keep the vehicle as far as possible on the center line of the lane and has achieved satisfactory results. However, this algorithm only considers the lateral control, not the longitudinal vehicle following. Lingli Yu et al. [15,16] proposed to use the DDPG algorithm to reduce the dependence on sample data. Their method had more continuous corner control and less lateral error when a vehicle was traveling. Although better results have been shown in the simulation environment, the agent is still affected by turn and throttle fluctuations and does not consider safety issues when interacting with other vehicles in highway conditions resulting in poor stability and safety.

To solve the above-mentioned problems, a vehicle safety planning and control method based on the variable Gauss safety field is designed in this paper. A planning model is constructed using a time series bird's-eye view as a state quantity and policy gradient algorithm. The timeliness and security of the planning model are verified by experiments. The reinforcement learning method of multi-task partitioning is used to partition and train the whole automatic driving trajectory tracking control task. Compared with the general end-to-end reinforcement learning auto-driving method, the multi-task partitioned training method reduces the training duration by dividing the entire auto-driving tracking control task into several sub-tasks and improves the noise input method in the longitudinal control module to further improve the training efficiency and provide a smoother driving experience. Meanwhile, protecting traffic participants is the most important topic in driving theory. Wang et al. [17,18] proposed the driving safety field theory modeling method and developed a collision warning algorithm, field experiments were conducted to verify the proposed algorithm. However, the whole framework contains several factors of driver, vehicle, and road, which bring great difficulties to practical application. To improve the practicability of safety field theory, a variable Gaussian safety field model is proposed to reveal the dynamic field characteristics of vertices. We use the variable Gaussian safety field model as the reward function of the planning module and combined with the constraint

and evaluation index of the control module. The model combines a Gaussian field in both directions to form an envelope and varies with the vehicle speed angle. While ensuring reasonable trajectory generation, the interaction of the ego vehicle with the surrounding vehicles is utilized to actively avoid the surrounding vehicles when they enter the Gaussian field, which improves the safety performance of the vehicle in high-speed scenarios such as highways. The simulation results in CARLA show that the vehicle safety planning control method based on the variable Gauss safety field has good planning efficiency and better safety compared with the traditional algorithms.

The main contributions of this paper are as follows:

(1) An automatic driving trajectory-planning method based on time series bird's-eye view and policy gradient algorithm is designed. The policy gradient algorithm is used to improve the ability of automatic driving vehicle trajectory planning and the efficiency of Lattice sampling method for trajectory planning. The time series bird's-eye view combined with the policy gradient algorithm can enhance the ability of feature extraction of the policy network, make the network convergence easier, and improve the feasibility of the method.

(2) The variable Gauss security field is added as the evaluation index of the reward function and control part to improve the security of trajectory and control effect.

2. Route Planning Algorithm

The goal of trajectory planning for autonomous driving is to find the optimal trajectory in advance for a vehicle. On the one hand, it is necessary to ensure the safety of the vehicle; On the other hand, getting to the destination through obstacles as soon as possible, reducing traffic pressure and improving driving efficiency are also important criteria to measure the effectiveness of the planned trajectory. Figure 1 shows that the trajectory planning module plays a key role in the overall auto-driving system.

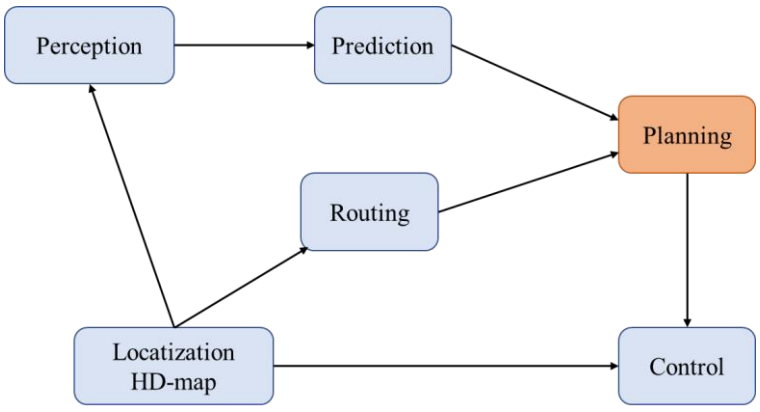

Figure 1. Autopilot system flow chart.

2.1. Time Series Bird's-Eye View and Strategic Network

The agents of reinforcement learning obtain the state input through interaction with the surrounding complex traffic environment to conduct effective learning training. One of the difficulties of the existing reinforcement learning algorithm is obtaining effective state features from complex environments. Overly redundant states will increase the learning difficulty of the agent. It is particularly important to make it easier for an agent to extract valid features. Therefore, this paper designs a policy network and corresponding time series bird's-eye view as the state quantity of the reinforcement learning, enabling the network to extract better environmental features.

2.1.1. Policy Network State Quantity

For an effective policy network for reinforcement learning, it is essential to obtain the perceptual information including lane lines, pedestrians, vehicles, and obstacles from

the surrounding environment as well as the predictive tracks for the next few moments including dynamic obstacles.

The sequential bird's-eye view significantly improves the learning efficiency of the policy network. Figure 2 shows the time series bird's-eye view matrix diagram.

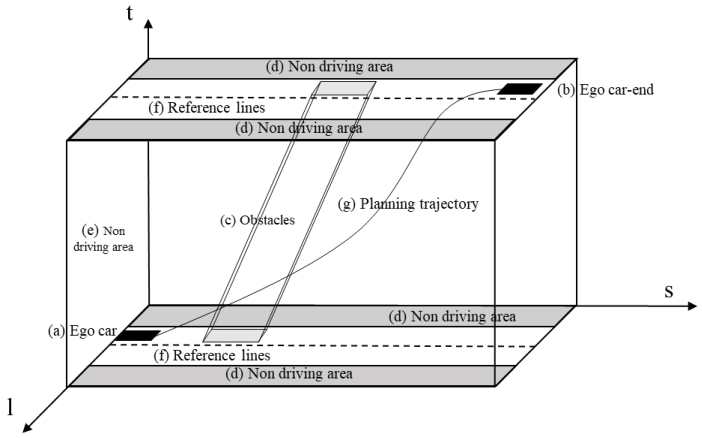

Figure 2. A time series bird's-eye view matrix diagram.

The bird's-eye view is a three-dimensional matrix composed of lateral displacement, vertical displacement and time. The specific elements in the matrix diagram shown in Figure 2 include (a) the current position status of the ego vehicle, (b) the ego vehicle, (c) obstacles, (d) the non-driving area and (e) the exercisable area, (f) the reference line, (g) the planned trajectory.

The generation of the time series bird's-eye view includes the following two steps: (1) According to the perception module of the autonomous vehicle, obtain the surrounding environmental information, including dynamic and static obstacles and lane lines. The prediction module is used to obtain the position information of dynamic obstacles in the future time of $0 \sim t_{end}$. (2) The information obtained from the perception module and the information is used to generate a bird's-eye view of features in three dimensions: horizontal, vertical and time.

The size of the three-dimensional the time series bird's-eye view matrix is (40, 400, 80). The first dimension 40 represents the horizontal range of 10 m on the left and right of the reference line, with the horizontal displacement interval of 0.5 m; The second dimension 400 represents the longitudinal 200 m forward range with the ego vehicle as the origin, the longitudinal displacement interval is 0.5 m, and the third dimension 80 represents the time range within the next 8 s, the time interval is 1 s. The (c) obstacles and (d) the non-driving area are represented by -1 in the time series bird's-eye view matrix; (e) the exercisable area is represented by 0 in the time series bird's-eye view matrix; (f) the reference line is represented by 1 in the time series bird's-eye view. In the matrix, the reference line represents higher priority than (c) obstacles, (d) the non-driving area and (e) the exercisable area. At the same time, (a) the current position status of the ego vehicle, (b) the ego vehicle, and (g) the planned trajectory are not specifically represented in the time series bird's-eye view matrix.

Figure 3 shows the vertical view of a time series bird's-eye view with a green rectangle representing the vehicles on the highway and a dashed grey line representing the driveway sidelines.

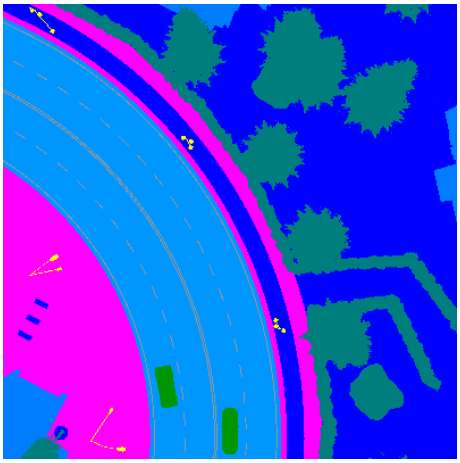

Figure 3. Time series bird's-eye view top view.

The generation of a time series bird's-eye view includes the following two steps: (1) Obtain the surrounding environment information, including dynamic and static obstacles, and lane lines, according to the perception module of the automobile. Obtain dynamic obstacles using prediction module in the future $0 \sim t_{end}$ location information within the end. (2) Generate cross-sectional, vertical, and temporal feature bird's-eye views using the information obtained from the perception and prediction modules. Then, train using the bird's-eye view as the state input.

2.1.2. Strategic Network Structure

Figure 4 shows the structure of the policy network $\pi_\theta(z, a)$. The network includes a convolution feature extraction network consisting of one convolution layer and a fully connected network consisting of three fully connected layers. Where z is the input state quantity of the policy network, including the time series bird's-eye view matrix and the history track of the vehicle, θ denote the weights and offset parameters for the network and a is the output of the policy network, that is, the final state of the planning trajectory $a = \left\{s, \dot{s}, \ddot{s}, l, \dot{l}, \ddot{l}, t\right\}$, where s, $\dot{s}$ and $\ddot{s}$ are the final longitudinal position, the end-of-longitudinal speed, and the acceleration of the longitudinal end state of the vehicle, respectively, while l, $\dot{l}$ and $\ddot{l}$ are the lateral end state position, the lateral end-state speed and the acceleration of the lateral end state of the vehicle, respectively. The input of the convolution feature extraction network is the time series aerial view matrix and the output is the final extracted environmental feature information. The input of the fully connected network is the convolution feature. The environmental feature information and the historical track information of the vehicle are extracted from the network output.

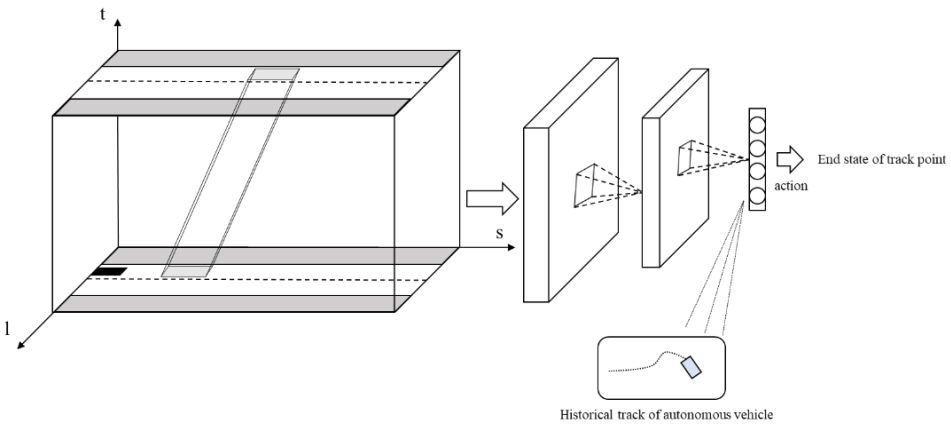

Figure 4. Strategic network structure diagram.

2.2. Variable Gauss Safety Field Theory

Since reinforcement learning explores policies and rewards by making agents constantly try and error, the security of reinforcement learning is lower than the other methods. Improving the security of reinforcement learning remains the focus of research. The variable Gauss security field model based on risk center transfer further improves the security of trajectory planning and control methods and serves as the reward function of the trajectory planning part and the constraint boundary of the control part.

Figure 5 shows that a static vehicle is abstracted as a rectangle with a length of l_v, a width of w_v, and the risk center $O(x_0, y_0)$ is its geometric center. The static security field of the vehicle is described by a two-dimensional Gaussian function as:

$$S_{sta} = C_a \cdot \exp\left(-\frac{(x-x_0)^2}{a_x^2} - \frac{(y-y_0)^2}{b_y^2}\right) \tag{1}$$

where C_a is the field strength factor, a_x and b_y represent the function of vehicle shape. The main control parameter for the shape of a static safety field is anisotropy:

$$\varepsilon = \frac{a_x^2 - b_y^2}{a_x^2 + b_y^2} = \frac{\phi^2 - 1}{\phi^2 + 1} \tag{2}$$

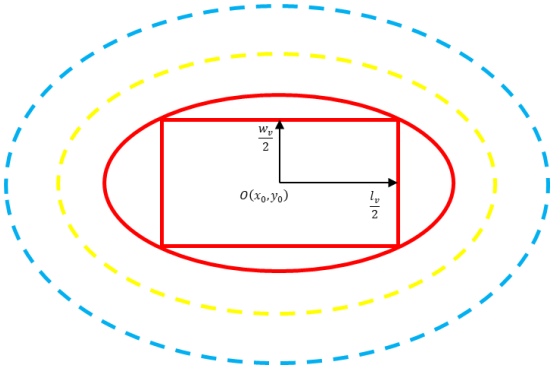

Figure 5. Static safety field overhead projection.

Parameter ε equivalently expressed in aspect ratio $\varnothing = a_x/b_y = l_v/w_v$.

The direction of the safety field is a vector from the risk center whose isoelectric line is projected upward into a series of ellipses. In Figure 5, the red rectangle represents the vehicle, the area in the solid red rectangle is called the core domain, the area between the red and the yellow ellipses is called the restriction domain, the area between the yellow and the blue ellipses is called the expansion domain, and each area represents a different risk state. The sizes of these different domains are related to the shape and motion of the vehicle and can be determined based on the parameters a_x, b_y of the Gaussian function (1). The Gauss security field is variable. The aspect ratio of the virtual vehicle will change with the change of the vehicle motion state and will significantly change the core, restriction and extension domains of the Gauss security field.

Figure 6 shows the overhead projection of the dynamic safety field. It can be seen that when the vehicle is in motion, the risk center will transfer following the vector $k_v \vec{v}$, the new risk center becomes $O'(x_0', y_0')$ and there are:

$$\begin{cases} x_0' = x_0 + k_v \left|\vec{v}\right| \cos\beta \\ y_0' = y_0 + k_v \left|\vec{v}\right| \sin\beta \end{cases} \tag{3}$$

where $\vec{v}$ is the velocity vector of the vehicle motion, k_v is the regulator and $0 < k_v < 1$ or $-1 < k_v < 0$, the sign corresponds to the front and back directions of the movement. β is the transferred angle between the vector and the x-axis.

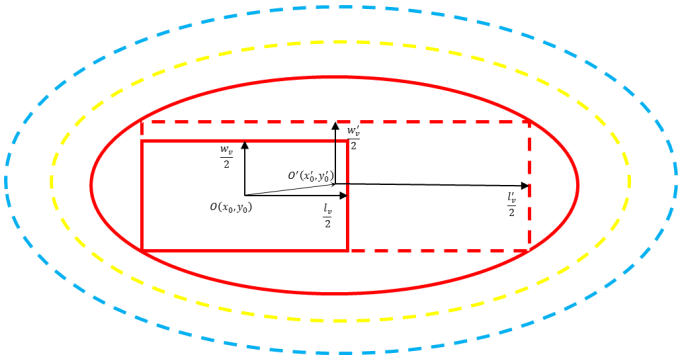

Figure 6. Overhead projection of dynamic safety field.

A virtual vehicle is formed with a length of l_v' and width of w_v' under the transfer of the risk center, whose geometric center is (x_0', y_0'), which establishes its dynamic security field as:

$$S_{dyn} = C_a \cdot \exp(-\frac{(x - x_0')^2}{(a_x')^2} - \frac{(y - y_0')^2}{(b_y')^2}) \quad (4)$$

where a_x' and b_y' are parameters related to vehicle shape and motion state. The new aspect ratio is expressed as $\varnothing' = a_x{}'/b_y{}' = l_v{}'/w_v{}'$.

2.3. Improved Lattice Programming Algorithm Based on Strategic Gradient Algorithm

The traditional Lattice programming algorithm achieves trajectory planning by sampling the target vertically and horizontally. This method will lead to the dilemma of a suboptimal solution for the sample-fitting trajectory, and it would be difficult to obtain the optimal trajectory. However, too many sampling points will lead to complex and inefficient calculations.

The Lattice algorithm is improved by using the policy gradient algorithm to directly obtain the optimal final state sample points as shown in Figure 7. This improved method abandons sampling with high time complexity and cost function evaluation for each alternate trajectory, which considerably improves the timeliness of the algorithm. Although the training process of reinforcement learning has better universality than the general rule-based planning algorithm, the design of the reward function based on the final control effect will make it more suitable for complex traffic scenes and complex vehicle dynamic features.

2.3.1. Track Planning Agent Design

The trajectory output by general dynamic programming, Monte Carlo sampling and time series difference methods will have a complete state action sequence $< s_0, a_0, s_1, a_1 \cdots s_{end-1}, a_{end-1}, s_{end} >$ and a trajectory consists of several state–action pairs as shown in Figure 8. Different actions a in each step will inevitably lead to changes in the overall trajectory. This will necessarily result in an exponential increase in the complexity of the solution as the length of the trajectory will increase. The simplified trajectory τ is composed of the start state s_0, action a and end state s_{end}. In the start state s_0, executing action a produces a unique trajectory τ, reaching the end state s_{end}.

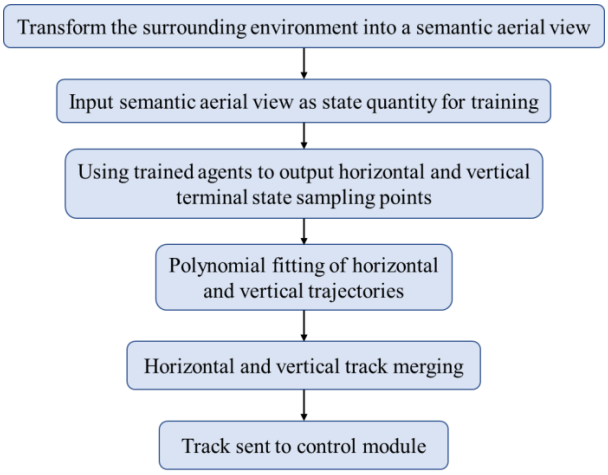

Figure 7. Lattice sampling process improvement.

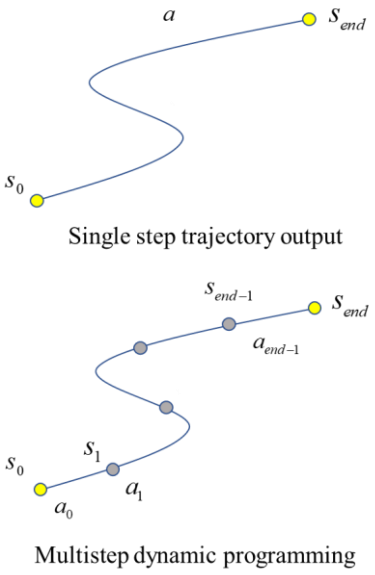

Figure 8. Diagrams of single-step and multi-step dynamic planning trajectory outputs.

In practice, the policy gradient algorithm is used instead of the last state sampling process in the Lattice algorithm. The end state of the track is used as the action space A:

$$A = \left\{ s_{end}, \dot{s}_{end}, \ddot{s}_{end}, l_{end}, \dot{l}_{end}, \ddot{l}_{end} \right\} \tag{5}$$

Policy network $\pi_\theta(z, a)$ maximizes the expected return of the output trajectory as an optimization objective:

$$J(\pi) = \sum_\tau p(\tau, \theta) \cdot r(\tau) \tag{6}$$

where z denotes the state features of the surrounding traffic environment, a is the network output action, θ is a network parameter, $p = (\tau, \theta)$ is the probability of executing action a and outputting track τ under parameter θ and state z, and $r(\tau)$ is the reward function of trajectory τ.

The gradient rise method is used to optimize $\pi_\theta(z, a)$ from Equation (6):

$$\theta = \theta + \alpha \cdot \nabla_\theta J(\pi) \tag{7}$$

To calculate the derivative of the optimization objective with respect to network parameter θ, the strategy gradient is derived as:

$$\begin{aligned} &\nabla_\theta J(\theta) = \nabla_\theta \sum_\tau p(\tau,\theta)\cdot r(\tau) \\ &= \sum_\tau \nabla_\theta p(\tau,\theta)\cdot r(\tau) \\ &= \sum_\tau \frac{p(\tau,\theta)}{p(\tau,\theta)} \nabla_\theta p(\tau,\theta)\cdot r(\tau) \\ &= \sum_\tau p(\tau,\theta)\nabla_\theta \log p(\tau,\theta)\cdot r(\tau) \end{aligned} \tag{8}$$

To improve the efficiency of training, during the training process, the agent continuously stores the experience data $< z, a, \tau, r >$ from the interaction with the environment in real-time into the experience pool (Memory). The Monte Carlo method is also used to randomly extract the mini-batch-sized empirical data from the experience pool for training:

$$\nabla_\theta J(\theta) \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \log p(\tau,\theta)\cdot r(\tau) \tag{9}$$

From Formula (9), the update direction of the final policy parameters θ is:

$$\theta = \theta + \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \log p(\tau,\theta)\cdot r(\tau) \tag{10}$$

To enhance the agent's exploring ability in unfamiliar state space and avoid the agent falling into local optimal space during training, the output of the policy network $\pi_\theta(z,a)$ will conform to normal distribution. It consists of two parts: mean $\mu(z,a)$ and variance $\sigma(z,a)$:

$$\pi_\theta(z,a) = \frac{1}{\sqrt{2\pi}\cdot\sigma(z,\theta)} \exp\left(-\frac{(z-\mu(z,\theta))^2}{2\sigma^2(z,\theta)}\right) \tag{11}$$

During the learning process of the policy network $\pi_\theta(z,a)$, the mean $\mu(z,a)$ and the variance $\sigma(z,a)$ of the output keep approaching $arg_{max}Q(z,a)$ and 0, respectively, and the probability of the agent taking random behavior exploration keeps decreasing. During training, the agent selects action $a = \left\{s, \dot{s}, \ddot{s}, l, \dot{l}, \ddot{l}, t\right\}$ from this normal distribution as the training output and executes it.

2.3.2. Reward Function Design

Reinforcement learning obtains the amount of state by interacting with the environment and evaluates the training agent by a reward function. The agents obtain higher returns by continuously optimizing their network of policies. Therefore, the design of the reward function is critical to the convergence of the agent, which affects the final decision-making results of the overall model. Moreover, a reasonable reward function design can also make the agent obtain more incentives from the environment and accelerate the convergence speed of the agent.

The reward function design for the trajectory planning section includes the following sections:

$$reward = k_1 \cdot r_{speed} + k_2 \cdot r_{acc} + k_3 \cdot r_{lateral} + k_4 \cdot r_{comfort} + k_5 \cdot r_{additional} + k_6 \cdot r_{safe} \tag{12}$$

In the formula, $r_{speed} = -\sum_{t<t_{total}} t\cdot\left(v_{target} - v_t\right)^2$ is the speed reward, its goal is to keep the speed at the target speed; $r_{acc} = -\sum_{t<t_{total}} \ddot{s}_t^2$ and $r_{comfort} = -\sum_{t<t_{total}} \ddot{l}_t{}^2$ are the longitudinal and lateral comfort rewards, respectively, their goals are to maintain low longitudinal acceleration and low lateral acceleration, respectively; $r_{lateral} = -\sum_{t<t_{total}} l_t^2$ is the lateral deviation reward, its goal is to maintain a small lateral deviation from the reference line; $r_{additoanal} = -\sum_{t<total}\left(s_t - s_t^{actual}\right)^2 + \left(l_t - l_t^{actual}\right)^2$ is the additional coupling reward, the objective is to maintain the coupling force between the planned trajectory and the controller and vehicle dynamics, and to maintain a better horizontal and vertical track-

ing accuracy of the vehicle during actual tracking; and r_{safe} is the safety reward. $k_1 \sim k_6$ is the proportion weight of each reward function. Where, $k_1 = 1.0$, $k_2 = 0.2$, $k_3 = 1.0$, $k_4 = 0.2$, $k_5 = 0.5$, and $k_6 = 1.0$. The value of $k_1 \sim k_6$ is obtained through debugging, and the specific value comparison is shown in Figure 9 below.

k1	k2	k3	k4	k5	k6	Mean-reward
1.0	0.2	1.0	0.2	0.5	1.0	0.829
1.0	0.2	0.5	0.2	0.5	1.0	0.647
1.0	0.1	1.0	0.1	0.5	1.0	0.532
1.0	0.5	1.0	0.5	0.5	1.0	0.247
1.0	0.2	1.0	0.2	0.8	1.0	0.073
1.0	0.1	1.0	0.1	0.5	1.0	0.532
1.0	0.2	1.0	0.2	0.2	1.0	−0.314
1.0	0.2	1.0	0.2	0.5	0.5	−2.012
1.0	0.2	0.5	0.2	0.5	0.5	−4.829

Figure 9. Comparison chart of proportional weights.

The design of r_{safe} is constrained by the variable Gaussian safety field, as shown below: When the vehicle is stationary:

$$\begin{cases} r_{safe,lat} = \begin{cases} -100 & if\ d_{lat} < l_v \\ 0 & if\ d_{lat} > l_v \end{cases} \\ r_{safe,lon} = \begin{cases} -100 & if\ d_{lon} < \frac{3l_v}{\sqrt{9-w_v^2}} \\ 0 & if\ d_{lon} > \frac{3l_v}{\sqrt{9-w_v^2}} \end{cases} \end{cases} \tag{13}$$

When the vehicle is moving:

$$\begin{cases} r_{safe,lat} = \begin{cases} -100 & if\ d_{lat} < l'_v \\ 0 & if\ d_{lat} > l'_v \end{cases} \\ r_{safe,lon} = \begin{cases} -100 & if\ d_{lon} < \frac{4l'_v}{\sqrt{16-(w'_v)^2}} \\ 0 & if\ d_{lon} > \frac{4l'_v}{\sqrt{16-(w'_v)^2}} \end{cases} \end{cases} \tag{14}$$

where $\begin{cases} w'_v = w_v + 2{\cdot}k_v{\cdot}\left|\vec{v}\right|{\cdot}\sin\beta \\ l'_v = l_v + 2{\cdot}k_v{\cdot}\left|\vec{v}\right|{\cdot}\cos\beta \end{cases}$, l_v and w_v are the length and the width of the agent, respectively, $\vec{v}$ is the speed vector of vehicle motion, k_v is the adjustment factor, and β is the angle between the transfer vector and the x-axis. After the actual vehicle test, k_v = 0.35.

3. Controller Design

The traditional trajectory planning module and the control module are simple upper and lower-level relationships. The trajectory planning module outputs the optimal trajectory and the controller tracks the control. Although this mode is simple and easy to operate, it cannot meet the real-time requirements in complex traffic environments. Figure 10 shows the relationship diagram of the proposed feedback design model. It can be seen from the

figure that the trajectory planning agent based on the policy gradient algorithm, the trajectory tracking controller and the environment form a planning control environment closed loop. The proposed loop feedback design model will enable the agents to continuously learn to adapt to the environment and adapt to the trajectory tracking controller. This method effectively links the traffic environment, the planner and the controller, so that the output trajectory of the planner can effectively adapt to the dynamic features of the vehicle and the controller. To enable the agent to stably, efficiently and safely track the optimal trajectory output by the planner, and improve the efficiency, the training of the control part is divided into horizontal control and vertical control.

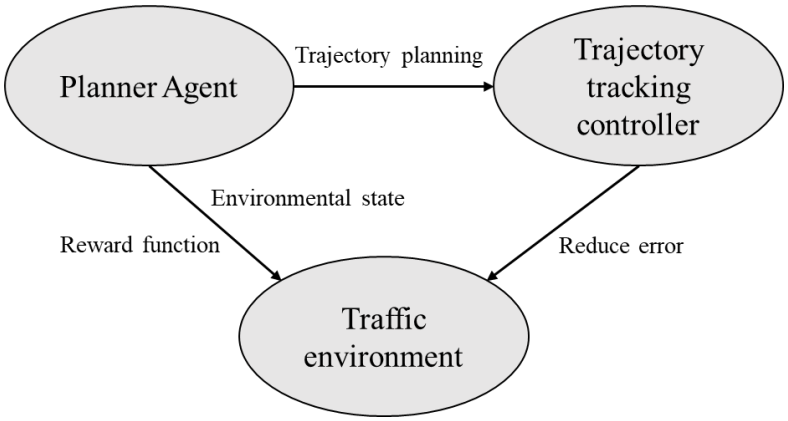

Figure 10. Planner/Controller/Environment relationship diagram.

3.1. Horizontal Trajectory Tracking Control Model Training

The goal of the traditional horizontal trajectory tracking task [19,20] is to enable vehicles to drive stably on the lane line without deviating, regardless of the state relationship with other vehicles. However, when the vehicle tracks and controls the track, the first consideration is the safety of the track, that is, it will not collide with other vehicles. Therefore, the variable Gaussian safety field is introduced as the evaluation index, and the state quantity and reward function are adjusted. The variables including the distance d_i from other vehicles, the lateral relative coordinate x_{i-v}, the coordinate (x_i, y_i) of the navigation point in the current vehicle coordinate system, the heading deviation φ and the speed v and acceleration $\dot{v}$ of the control vehicle are added as the state variables:

$$s^{lane-keep} =< d_0, d_1, \ldots, x_0, x_1, \ldots, x_{1-v}, x_{2-v}, \ldots, \varphi, v, \dot{v} > \tag{15}$$

The output action is only the steering wheel angle $a_{steer} \in [-1, 1]$. For the design of the reward function for lane keeping, the lateral error x_0 between the current vehicle coordinate and the lane centerline, the deviation φ of the heading angle and the relative distance d_i from other vehicles are considered as the evaluation index reward functions:

$$\begin{cases} r_{safe,lat} = -\log(\left|\frac{1}{\sqrt{2}} w'_v - d\right| + 1, 1.2) \\ r_{safe,lon} = -\log(\left|\frac{1}{\sqrt{2}} l'_v - d\right| + 1, 1.2) \\ r^{lane-keep} = -k_1 abs(x_0) - k_2 \sin\varphi \end{cases} \tag{16}$$

where $\begin{cases} w'_v = w_v + 2 \cdot k_v \cdot \left|\vec{v}\right| \cdot \sin\beta \\ l'_v = l_v + 2 \cdot k_v \cdot \left|\vec{v}\right| \cdot \cos\beta \end{cases}$, l_v and w_v are the length and the width of the agent, respectively, $\vec{v}$ is the speed vector of vehicle motion, k_v is the adjustment factor, and β is the angle between the transfer vector and the x-axis. After the actual vehicle test, k_v = 0.35.

If the lateral deviation of the current position of the autonomous vehicle is greater than the set maximum lateral deviation threshold value x_{0max} during the training, the

current round of iterative training will be ended for the next round of training. Through the cumulative reward mechanism, agents that enhance learning continuously obtain higher reward reports. Hence, they can take more potential threats into account. However, the dynamic features of the vehicle will be hidden in the state quantity of the past few moments. Thus, it would be difficult to fully understand the current state of the intelligent vehicle only through the current state quantity. To enable the agent to better understand the dynamic features of the intelligent vehicle at the current time and output more reasonable trajectory tracking actions, the state quantities at the current time and at the past four times are stacked together as network inputs.

3.2. Training of Longitudinal Trajectory Tracking Control Model

To maintain an ideal distance between the ego vehicle and the vehicle in front without any collision with the vehicle in front, the ego vehicle is expected to cruise at a constant speed when there is no vehicle in front. When there are other vehicles in front of the ego vehicle, the road information is not considered, instead only the information of the current vehicle and the vehicle ahead is considered as the state quantity. Figure 11 describes the cruise mission status. The longitudinal trajectory tracking control task considers the speed v and acceleration $\dot{v}$ of the current vehicle, speed v_l and acceleration $\dot{v}_l$ of the vehicle in front, the distance d from the vehicle in front and the expected speed v_{des} of the current vehicle as the state variables:

$$s^{acc} =< y_0, y_1, \ldots, \varphi, v, \dot{v} > \tag{17}$$

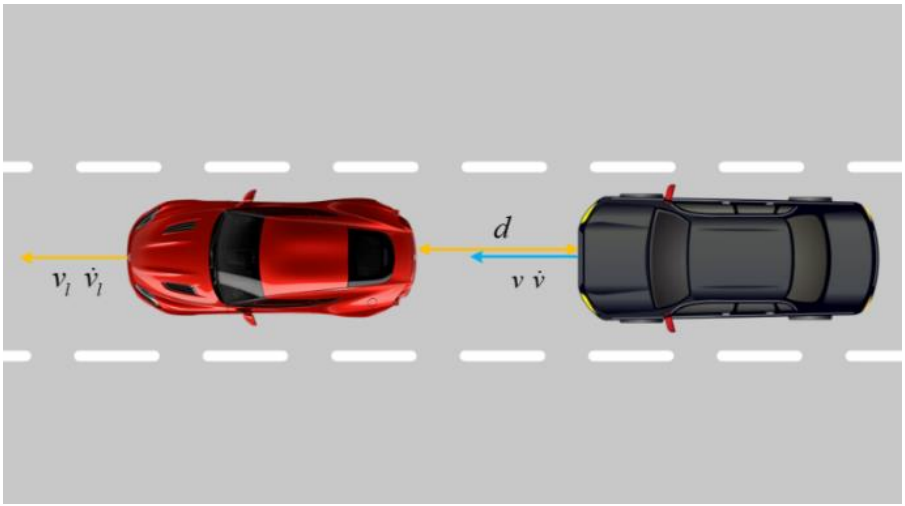

Figure 11. Description of cruise mission status.

Output action $a_{acc} \in [-1, 1]$ of the agent, including accelerator action $a_{throttle}$ and brake action a_{brake}:

$$\begin{cases} a_{throttle} = a_{acc}, a_{brake} = 0 & if\ a_{acc} \geq 0 \\ a_{throttle} = 0, a_{brake} = a_{acc} & if\ a_{acc} < 0 \end{cases} \tag{18}$$

For vertical control tasks, the reward function is designed as:

$$r^{acc} = \begin{cases} -k_5 abs(v - v_{des}) - k_6 abs(d - d_{des}) & if\ d > d_{safe} \\ -100 & if\ d \leq d_{safe} \end{cases} \tag{19}$$

where d_{des} and d_{safe} are the expected and safe distances from the vehicle in front, respectively. When the distance between the intelligent vehicle and the vehicle in front is less than the safe distance, the reward is -100 and the current interaction is stopped to start the next round of interaction. During longitudinal training, the speed v_l of the vehicle in front

and the expected speed v_{des} of the current vehicle are randomly given each round, so that the training model can be generalized to more complex situations.

The traditional training mostly uses Gaussian noise or Ornstein Uhlenbeck (OU) noise to promote agents to actively explore the environment at the beginning of training. However, unnecessary exploration will prolong the training time of agents. Therefore, in this paper, a Multi-Head Actor network structure is designed for the tasks with convex solution space in longitudinal control tasks. The main function of the proposed structure is to make the output action noisy. Action noise reflects the uncertainty measure of the optimal solution of the current policy. The Multi-head Actor network structure is used to construct this uncertainty measurement method.

The output of the Online Actor network is connected to multiple Head networks. To reflect the difference of each Head network, the initialization and training sampling experience pool of each Head network are independent and the way to converge to the optimal solution space is also different. Therefore, the variance of the Head network output action is used to estimate the uncertainty measure of the output action of the Actor network as:

$$\begin{cases} N_t = \{k \cdot \mathrm{var}(\mu(s_t|\theta_{\mu online})) & if\ k \cdot \mathrm{var}(\mu_{\theta_{\mu_{online}}}(s_t|\theta_{\mu online})) < N_{threshold} \\ N_{threshold} & else \end{cases} \tag{20}$$

where N_t and $N_{threshold}$ are the real-time action noise and the threshold noise, respectively, θ is the adopted policy, $\mu\left(s_t|\theta_{\mu online}\right)$ is the deterministic action of the network output, and k is the weight parameter.

Similar to the horizontal control part, the vertical control part also selects the current state quantity of the agent and the state quantity of the past four times as the network input, making the network easier to converge and having high training efficiency.

4. Experiment and Analysis

The simulation experiment is based on the open-source autopilot simulator CARLA, which supports the development, training and validation of autopilot systems. In addition to open-source code and API protocol, CARLA also provides open mathematical assets (urban layout, buildings and vehicles) that can be freely invoked. CARLA works through the client mode. It has a specific python API interface that can realize simulation environment configuration, environment interaction and vehicle control through interface code. CARLA is suitable as a training platform for automatic driving reinforcement learning. The simulation training was completed under the environment of TOWN06 and TOWN04 in CARLA 0.9.9. Figure 12 shows the specific CARLA simulation scenario.

Figure 12. CARLA simulation diagram.

4.1. Trajectory Planning Experiment Based on PG Algorithm

When training the trajectory planning module, other obstacle vehicles were randomly generated for each round of training to enable the trained agents to target complex traffic conditions. In a random environment, the average reward of each round was used to evaluate the training effect of the agent. When the agent reached the specified number of steps or encounters a collision, it directly started the next round of training. To avoid

randomness, the final training results were obtained by averaging the five training results. The training results are shown in Figure 13. The red curve is the average reward, and the red-shaded part is the sliding average of the five training rewards. Due to the strong randomness of the training environment, the rewards show a strong jitter with the change of the round. The rewards show an overall upward trend with the change of rounds, indicating that the agents are increasingly adapting to the changing traffic environment to obtain higher rewards during the training process. After 100 rounds, the variance of rewards tends to decrease, and the training results of agents become more stable.

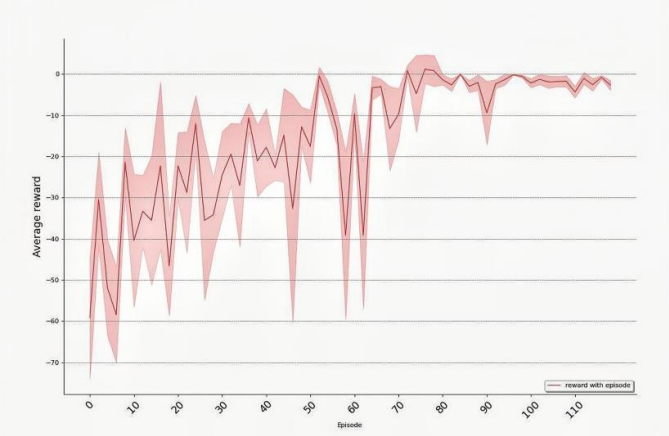

Figure 13. The change in average reward of the track planning module with the number of training wheels.

As shown in Figure 14, the red curve represents the reward curve of the planning method based on the time series bird's-eye view and the policy gradient algorithm proposed in this paper, and the blue curve represents the reward curve of the planning method using the DDPG algorithm. Because of the strong randomness of the training environment, the reward fluctuates greatly with the change of the round. In the comparison of average rewards, both curves are almost the same. However, it is obvious that the DDPG algorithm represented by the blue curve has convergence effect only after 100 rounds, while the planning method proposed in this paper starts to converge gradually after 70 rounds. Therefore, the proposed planning method has higher convergence efficiency and stability.

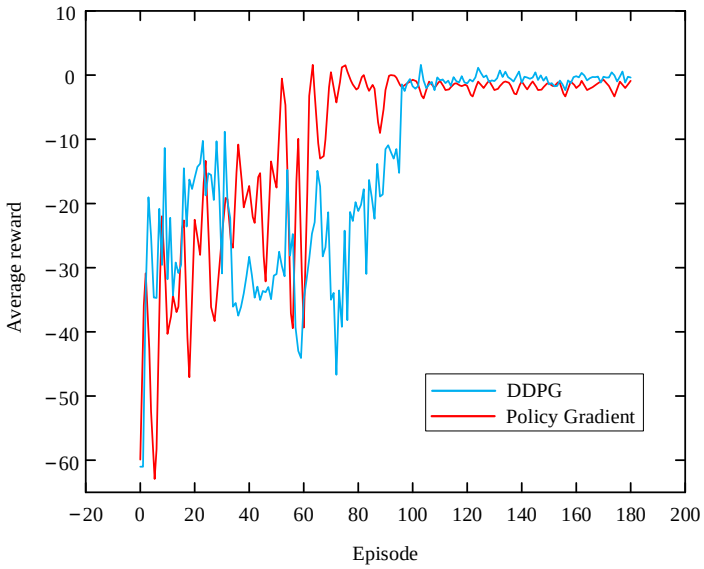

Figure 14. Comparison curve of average reward of the track planning module changing with the number of training wheels.

4.2. Safety Control Module Experiment

In the control module, due to the randomness of the steps that the autonomous vehicle can take during the training process, it is not suitable to use a single reward or a cumulative reward as the evaluation standard of the training effect of the agent at the current moment. Therefore, it is reasonable to take the average reward of each step of the current round as the evaluation standard of the training effect of the current round. The abscissa is the number of training rounds, and the ordinate is the average reward obtained in each round. Figure 15 shows the change in the training curve of the horizontal trajectory tracking task.

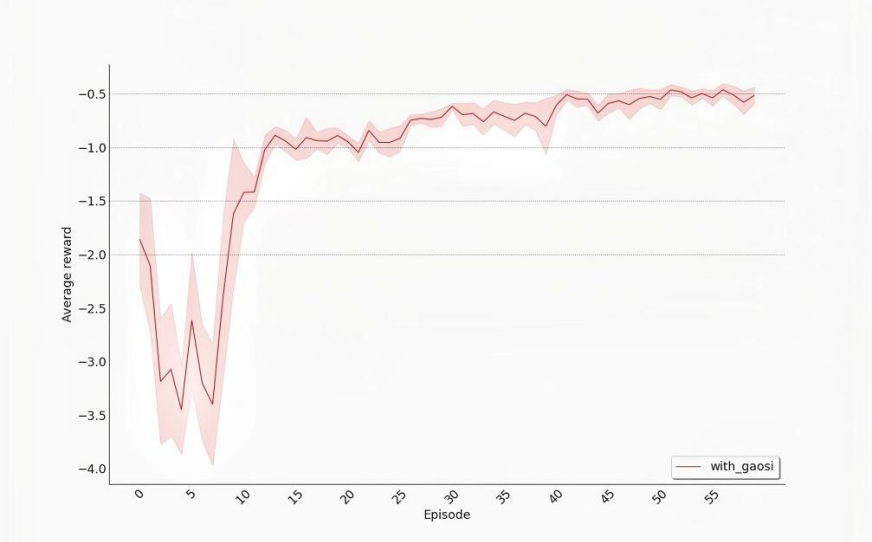

Figure 15. The change in average reward of the horizontal trajectory tracking task with the number of training wheels.

It can be seen from Figure 15 that in the first 15 rounds of the lateral trajectory tracking control task, the agent is still in the free exploration stage, and the reward curve fluctuates and does not converge. With the progress of training, the agent continuously optimizes its strategic network, makes more reasonable behavior, obtains higher rewards and optimizes its network again according to the rewards obtained from feedback, forming a virtuous circle. After 50 rounds, the reward curve begins to converge and achieves good training results.

In this paper, the variable Gaussian safety field is used as the constraint and evaluation index of the control part. Figure 16 shows the reward curve of the variable Gaussian safety field. The red curve represents the reward curve of the lateral tracking control considering the relationship with other vehicle state quantities under the variable Gaussian safety field. The blue curve represents the reward curve of the traditional lateral tracking control under the variable Gaussian safety field. In both cases, the average value of the five experiments is taken. Figure 16 clearly shows that the reward curve of the safety lateral tracking control method proposed in this paper is superior to the traditional lateral tracking control, with higher safety performance and greater response space to emergency conditions. At the beginning of several training rounds, since the agent did not interact with other vehicles in the opening exploration phase, the average reward was 0, as shown in Figure 16. From the sixth round, the agent interacts with other vehicles in the environment, the variable Gaussian safety field acts, and the reward curve changes.

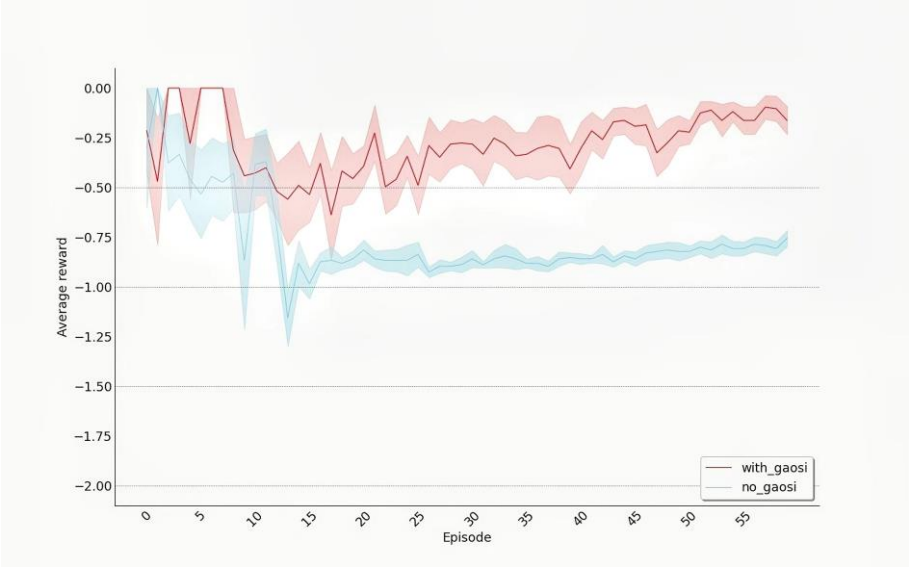

Figure 16. Reward curve of variable Gauss safety field.

Figure 17 shows the average reward of the longitudinal trajectory tracking control task over time. It can be seen that the average reward changes with the training times. The blue and red curves represent the average reward change curves of the agents with Gaussian noise and adaptive noise exploration, respectively, and the shaded part is the standard deviation of five experiments. Figure 17 shows that both types of agents have achieved good training results in the longitudinal trajectory tracking control task. Due to the randomness of the ego vehicle's speed and the state of the vehicle ahead in each training round, the average reward of the lateral trajectory tracking control task fluctuates to some extent. However, similar to the lateral trajectory tracking control task, the training effect of the adaptive noise detection method is better than that of the common noise attenuation method.

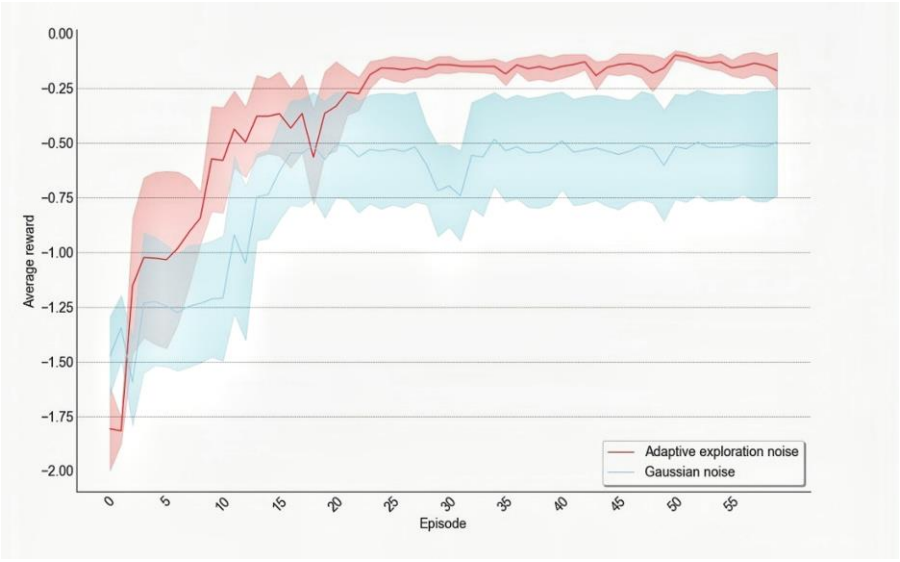

Figure 17. The change in the average reward of the longitudinal trajectory tracking task with the number of training wheels.

5. Conclusions

In this paper, a vehicle safety planning control method based on the variable Gaussian safety field is designed. The policy gradient algorithm is used to improve the driving safety

of autonomous vehicles and make the driving trajectory of autonomous vehicles more intelligent. The spatiotemporal bird's-eye view proposed in combination with the policy gradient algorithm as a state variable can enhance the ability of feature extraction of the policy network and make the network convergence easier. The variable Gaussian safety field is added as the reward function of the trajectory planning module and the evaluation index of the control module to improve the safety and rationality of the output trajectory and tracking control, respectively. In the longitudinal control module, Gaussian noise input is improved to avoid repeated invalid exploration of agents and enhance training efficiency. Compared with the traditional planning control algorithm, the proposed method has the following advantages: (1) the spatiotemporal bird's-eye view is used as the input state of the policy network enabling the trajectory planning policy network to effectively extract the features of the surrounding traffic environment. The planning trajectory of autonomous vehicles is generated through reinforcement learning, which improves the trajectory planning ability of autonomous vehicles in complex scenes. The efficiency of the lattice sampling method for trajectory planning algorithm avoids invalid sampling in complex traffic scenes; (2) the variable Gaussian safety field is added as a reward function to improve the safety of trajectory and control effect; (3) the traditional noise input is improved and the multi-head actor network structure is designed to add noise in the output action and improve the training efficiency. The experimental results demonstrate and validate that the proposed framework is superior to the traditional methods.

At the same time, this paper does not consider the scenarios other than an expressway, and how to change lanes in an emergency. In the future, we will test and improve the algorithm in more complex environments, such as ramps and urban roads. From another point of view, the single vehicle will be extended to the fleet, and the driving efficiency and safety of the fleet on the expressway will be considered.

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

Funding: This research was funded by [National Natural Science Foundation of China] grant number [U20A20333, 51875255, U20A20331, 52072160] and [Six talent peaks project in Jiangsu Province] grant number [2018-TD-GDZB-022] and [Key R&D projects in Jiangsu Province] grant number [BE2020083-3, BE2019010-2].

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. And Yubo Lian is employee of BYD Company Limited. The paper reflects the views of the scientists, and not the company.

References

1. Hu, X.; Li, Z. An improved potential field method for robot path planning. *Mech. Sci. Technol. Aerosp. Eng.* **2017**, *36*, 1522–1529.
2. Fu, H.; Nie, W.; Wang, K.; Vagale, A.; Robin, T.B. Simulation and verification of path planning for autonomous vehicles based on sampling. *Auto Electr. Parts* **2021**, *9*, 13–15.
3. Chen, D.; Liu, X.; Liu, S. Improved A* algorithm based on two-way search for path planning of automated guided vehicle. *J. Comput. Appl.* **2021**, *41* (Suppl. S2), 309–313.
4. Peng, Y.; Liang, J. Q-learning path planning based on exploration /exploitation tradeoff optimization. *Comput. Technol. Dev.* **2022**, *32*, 1–7.
5. Qi, X.; Huang, J.; Cao, J. Path planning for unmanned vehicle based on improved A* algorithm. *J. Comput. Appl.* **2020**, *40*, 2021–2027.
6. Claussmann, L.; Revilloud, M.; Gruyer, D.; Glaser, S. A review of motion planning for highway autonomous driving. *IEEE Intell. Transp. Syst.* **2019**, *21*, 1826–1848. [CrossRef]
7. Li, X.; Sun, Z.; Zhu, Q.; Liu, D. A unified approach to local trajectory planning and control for autonomous driving along a reference path. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Hongkong, China, 3–6 August 2014.

8. Yu, C.; Cherfaoui, V.; Bonnifait, P. Semantic evidential lane grids with prior maps for autonomous navigation. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016.
9. Werling, M.; Ziegler, J.; Kammel, S.; Thrun, S. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3–8 May 2010.
10. Fehér, Á.; Aradi, S.; Hegedüs, F.; Bécsi, T.; Gáspár, P. Hybrid DDPG approach for vehicle motion planning. In Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Sapporo, Japan, 29–31 July 2019.
11. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A.A.; Yogamani, S.; Perez, P. Deep reinforcement learning framework for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4909–4926. [CrossRef]
12. Roy, A.; Hossain, M.; Muromachi, Y. A deep reinforcement learning-based intelligent intervention framework for real-time proactive road safety management. *Accid. Anal. Prev.* **2022**, *165*, 106512. [CrossRef] [PubMed]
13. Aradi, S.; Becsi, T.; Gaspar, P. Policy gradient-based reinforcement learning approach for autonomous highway driving. In Proceedings of the 2018 IEEE Conference on Control Technology and Applications (CCTA), Copenhagen, Denmark, 21–24 August 2018.
14. Nageshrao, S.; Tseng, E.; Filev, D. Autonomous highway driving using deep reinforcement learning. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019.
15. Yu, L.; Shao, X.; Wei, Y.; Zhou, K. Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning. *Sensors* **2018**, *18*, 2905. [CrossRef] [PubMed]
16. Jaritz, M.; De Charette, R.; Toromanoff, M.; Perot, E.; Nashashibi, F. End-to-end race driving with deep reinforcement learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018.
17. Wang, J.; Wu, J.; Li, Y. The driving safety field based on driver-vehicle-road interactions. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2203–2214. [CrossRef]
18. Wang, J.; Wu, J.; Zheng, X.; Ni, D.; Li, K. Driving safety field theory modeling and its application in pre-collision warning system. *Transp. Res. Part C* **2016**, *72*, 306–324. [CrossRef]
19. Zhe, X.; Hailiang, C.; Ziyu, L.; Enxin, S.; Qi, S.; Shengbo, L. Lateral trajectory following for automated vehicles at handling limits. *J. Mech. Eng.* **2020**, *56*, 138–145. [CrossRef]
20. Chen, H.; Chen, S.; Gong, J. A review on the research of lateral control for intelligent vehicles. *Acta Armamentarii* **2017**, *38*, 1203–1214.