electronics

MDPI

Article

VN-MADDPG: A Variable-Noise-Based Multi-Agent Reinforcement Learning Algorithm for Autonomous Vehicles at Unsignalized Intersections

Hao Zhang, Yu Du *, Shixin Zhao, Ying Yuan and Qiuqi Gao

Beijing Key Laboratory of Information Service Engineering, College of Robotics, Beijing Union University, Beijing 100101, China; 20221083510912@buu.edu.cn (H.Z.); 20221083510922@buu.edu.cn (S.Z.); 20221083510910@buu.edu.cn (Y.Y.); 20231083510904@buu.edu.cn (Q.G.)

* Correspondence: duyu@buu.edu.cn

Abstract: The decision-making performance of autonomous vehicles tends to be unstable at unsignalized intersections, making it difficult for them to make optimal decisions. We propose a decision-making model based on the Variable-Noise Multi-Agent Deep Deterministic Policy Gradient (VN-MADDPG) algorithm to address these issues. The variable-noise mechanism reduces noise dynamically, enabling the agent to utilize the learned policy more effectively to complete tasks. This significantly improves the stability of the decision-making model in making optimal decisions. The importance sampling module addresses the inconsistency between outdated experience in the replay buffer and current environmental features. This enhances the model's learning efficiency and improves the robustness of the decision-making model. Experimental results on the CARLA simulation platform show that the success rate of decision making at unsignalized intersections by autonomous vehicles has significantly increased, and the pass time has been reduced. The decision-making model based on the VN-MADDPG algorithm demonstrates stable and excellent decision-making performance.

Keywords: multi-agent model; autonomous driving decision making; intersection scenarios; variable noise

Citation: Zhang, H.; Du, Y.; Zhao, S.; Yuan, Y.; Gao, Q. VN-MADDPG: A Variable-Noise-Based Multi-Agent Reinforcement Learning Algorithm for Autonomous Vehicles at Unsignalized Intersections. *Electronics* **2024**, *13*, 3180. https://doi.org/10.3390/electronics13163180

Academic Editor: Ali Riza Ekti

Received: 10 July 2024
Revised: 7 August 2024
Accepted: 10 August 2024
Published: 11 August 2024

1. Introduction

The acceleration of urbanization has exacerbated urban transportation problems, particularly at complex road intersections. Decision making and planning for autonomous driving at intersections constitute a complex problem [1]. Numerous researchers have devoted their time to developing algorithms for autonomous driving decision-making models in relation to intersections [2].

Intersections without traffic signals are more complex and prone to accidents. There are more potential conflict zones along the lanes without traffic signal management. Changes in vehicle driving behavior are also more likely to cause confusion. Dense traffic from various directions can block the intersection, causing collisions, congestion, and safety accidents [3,4]. According to the U.S. National Highway Traffic Safety Administration's fatality analysis report, more than a quarter of all fatal crashes in the United States occur at or in connection with intersections, with approximately 50% of those occurring at uncontrolled intersections [5].

Due to the lack of traffic signals or signs, drivers need to decide on their own whether, when, and how to enter and pass through an intersection. With the emergence of autonomous vehicles, this task is transferred to machine learning and artificial intelligence algorithms. Autonomous vehicles can obtain road and other vehicle status information through high-resolution cameras, LiDAR, and mmWave radar sensors. He et al. [6] proposed the DAMO-StreamNet framework. Li et al. [7] proposed the LongShortNet network. Lv et al. [8] developed a fusion architecture. These advancements aim to improve real-time

perception tasks, providing more accurate perception results to support more reliable autonomous driving decision-making algorithms. This is crucial for navigating complex intersection scenarios. In this context, considering the interactions between vehicles is particularly important: if driving is too conservative, failing to account for these interactions may lead to a deadlock, where vehicles might become stuck and never pass through the intersection, or if driving is too aggressive, it could result in collisions [9]. The continuous improvement of autonomous driving technology can improve traffic safety and increase traffic flow. It can also provide economic benefits, environmental protection, and social inclusiveness [10].

Interactions between vehicles at unsignalized intersections are highly complex. Current research on autonomous driving strategies at intersections mainly concentrates on developing algorithms based on motion prediction, threat estimation, and cooperative decision making [11]. These studies generally assume that all vehicles on the road are autonomous [12], but human behavior often exhibits subjective uncertainty. Autonomous vehicles (AVs) need to interact with human-driven vehicles (HDVs), which exhibit unpredictable behavior. Therefore, decision-making algorithms for autonomous vehicles at unsignalized intersections must be able to cope with dynamically changing conditions and unpredictable actions, which has always been a challenge in the field of autonomous driving.

Our research aims to enhance existing multi-agent reinforcement learning algorithms to better address these complex decision-making challenges. We improve the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm [13] by enhancing the replay buffer and introducing a variable-noise mechanism. These enhancements increase the stability and robustness of the decision-making process in dynamic environments and the unpredictable behaviors of surrounding vehicles. The proposed VN-MADDPG algorithm makes decision making at complex intersections more efficient and reliable.

The contributions of our paper are as follows:

- The autonomous driving decision-making problem at unsignalized intersections is defined as a collaborative multi-agent reinforcement learning (MARL) problem. Vehicle behavior in training scenarios is more uncertain. Training in such scenarios enhances the ability of autonomous vehicles to handle various emergent situations in intersections. This enhances the success rate and efficiency of decision making.
- A variable mechanism is integrated into the noise module of the VN-MADDPG algorithm. It can gradually reduce the noise based on the proportion of remaining training episodes. This encourages the agent to rely more on the strategies it has already learned to complete tasks, enhancing the robustness and stability of the final decision model.
- The experience replay buffer is designed with an importance sampling module. It focuses more on samples that significantly impact the learning process. VN-MADDPG can utilize experiences in the replay buffer more effectively to improve learning efficiency. This helps the algorithm quickly converge to superior policies, further enhancing the robustness of the model.

The rest of this paper is organized as follows. Section 2 provides an overview of recent related works. Section 3 introduces the VN-MADDPG model. The simulation environment and the experimental setup are described in Section 4. In Section 5, we analyze the experimental results. Finally, Section 6 concludes our work.

2. Related Works

2.1. Single-Agent Reinforcement Learning

Single-agent reinforcement learning (SARL) algorithms have been widely applied in the field of autonomous driving, including Q-learning [14], DQN [15], DDPG [16], PPO [17], etc., but they typically focus on optimizing the policy of a single agent. They lack the ability to handle the complex cooperation and competition relationships among multiple agents. SARL usually assumes that the driving behavior of other vehicles is fixed and follows certain rules [18]. However, drivers of other vehicles may adjust their decisions based

on observed driving behaviors, leading to uncertainty and unpredictability. Single-agent algorithms cannot effectively predict and adapt to the dynamic strategies of other vehicles. This limitation not only reduces the flexibility and adaptability of the algorithms but can also lead to suboptimal or even ineffective strategies.

Li et al. proposed a deep reinforcement learning approach for autonomous vehicles at intersections, using a convolutional neural network with deep deterministic policy gradients. This method simplifies decision making and reduces computational complexity but struggles with multi-vehicle interactions in dynamic traffic environments [19]. Gutierrez et al. introduced a deep reinforcement learning method with curriculum learning for intersection handling, enabling the agent to infer vehicle intentions and intersection types without prior information. However, it lacks effective management of dynamic interactions among multiple agents [20]. Xiao et al. developed a decision-making framework for autonomous ego vehicles using Soft Actor-Critic (SAC), featuring a mixed-attention network and an enhanced replay buffer. While it handles uncertainty in dynamic environments well, it does not consider the driving intentions of surrounding vehicles [21].

2.2. Multi-Agent Reinforcement Learning

To address these issues, many researchers are dedicated to researching multi-agent reinforcement learning (MARL) methods. MARL algorithms involve several agents, such as autonomous vehicles, learning simultaneously in a shared environment and adjusting their strategies continually. Although this instability disrupts the equilibrium of the environment and hinders the learning process, it better reflects complex real-world scenarios [22]. MARL algorithms address these challenges by introducing mechanisms for cooperation, competition, and communication strategies. This makes them more suitable for application in complex decision-making scenarios, such as intersections with many vehicles. Other advanced technologies also have the capacity to achieve better decision-making results and are worth researching [18,23].

There are many modes of MARL. Centralized learning exhibits poor scalability in large-scale agent environments and may lead to some agents learning negative strategies. Decentralized learning faces the challenge of environmental non-stationarity. The centralized training and decentralized execution mode is more feasible. The problem shifts to how to train independent strategies for each agent from a global perspective [24].

In recent years, MARL has been utilized to solve many multi-agent problems. such as autonomous decision making [25–28]. It has considerable research prospects. Guan et al. [29] proposed a reinforcement learning (RL) training algorithm named Model-Accelerated Proximal Policy Optimization (MA-PPO). It integrates a prior model into the Proximal Policy Optimization (PPO) algorithm to enhance the learning process in terms of sample efficiency. Antonio et al. [30] proposed a novel advanced autonomous intersection management method based on end-to-end multi-agent deep reinforcement learning. It enables collaborative control of autonomous vehicles at intersections, autonomously learning complex real-world traffic dynamics. Zhuang et al. [31] modeled the decision-making process at intersections as a model-free, fully cooperative multi-agent system. They developed a multi-agent policy optimization algorithm based on attention-based representation to make joint decisions, avoiding collisions between vehicles and efficiently navigating through intersections. However, the MAPPO algorithm learns policies through centralized advantage value sampling, which may encounter the issue of policy overfitting in multi-agent cooperation. This ultimately leads to some agents updating policies in suboptimal directions and hinders agents from exploring better trajectories [32].

Wu et al. [33] proposed the CoMADDPG algorithm for connected vehicles at unsignalized intersections. Compared to optimization-based methods, CoMADDPG significantly reduces the average travel time. This demonstrates its potential at unsignalized intersections. Hu et al. [34] proposed a model in which multiple agents continuously interact, capturing and learning the inherent uncertainty in human behavior. Liu et al. [35] proposed a new, efficient algorithm, MA-GA-DDPG, to address the decision-making problem for

connected autonomous vehicles at unsignalized intersections. The algorithm incorporates an attention mechanism and a safety monitoring module to improve traffic safety while considering the heterogeneity of human drivers.

Despite these advancements, several issues remain in MARL models. For example, the MAPPO algorithm's reliance on centralized advantage value sampling can lead to policy overfitting and suboptimal policy updates [32]. Additionally, while the MADDPG model has shown promise in mitigating collisions, the stability and robustness of these algorithms still require further investigation [36]. Addressing these challenges is crucial for enhancing the performance and reliability of multi-agent systems in dynamic and complex environments.

3. Methods

To address these issues, we propose the VN-MADDPG algorithm. In this section, the framework of our VN-MADDPG model is first outlined. Then, we provide a detailed description of the variable-noise mechanism and the importance sampling module, which are essential for improving the learning efficiency of the algorithm and enhancing the robustness of the decision-making model.

We formulate the decision-making problem of multi-vehicle autonomous driving in a mixed-traffic environment as a Markov decision process based on a collaborative multi-agent reinforcement learning algorithm. The Markov decision process is used for agents to learn strategies and supports agents in coordinating conflict decisions [37].

We use MADDPG as the baseline algorithm due to its suitability for handling mixed cooperative-competitive environments and continuous action spaces, which are crucial in autonomous vehicle decision-making systems. MADDPG [13] excels in these settings by allowing each agent to implement a DDPG algorithm, making it effective for complex multi-agent scenarios. However, the dynamic changes in the environment can diminish the relevance of experience samples in the experience pool, causing models to converge slowly. This slow convergence hampers the rapid acquisition of effective strategies. Therefore, enhancing the stability and robustness of MADDPG is essential to better adapt to dynamic environments and the unpredictable behaviors of surrounding vehicles.

3.1. VN-MADDPG

To enhance the stability and robustness of MADDPG and help the model learn better policies more quickly, we propose the VN-MADDPG algorithm. The structure and data flow of VN-MADDPG is shown in Figure 1.

In VN-MADDPG, the critic network of each agent is updated based on the policies of all agents, similar to MADDPG, while each agent's actor network focuses only on its own observations. This can provide superior policy learning performance and promote cooperation among multiple agents effectively, thus enabling them to pass through intersections smoothly.

VN-MADDPG also utilizes an experience replay buffer module. Each experience sample consists of $\{s, a_1 \ldots a_N, r_1 \ldots r_N, s^{'}, d\}$, where s denotes the observation vector of all agents within the current environment for the current environment state $\{s_1, s_2 \ldots s_n\}$ and s' denotes the observation vector of all agents within the current environment at the next time step. $\{a_1, a_2, \ldots, a_N\}$ is the action set of the agents, and$\{r_1, r_2, \ldots, r_N\}$ is the set of rewards of the agent. d is a boolean value indicating whether all trained agents in this round arrived at the destination without collision.

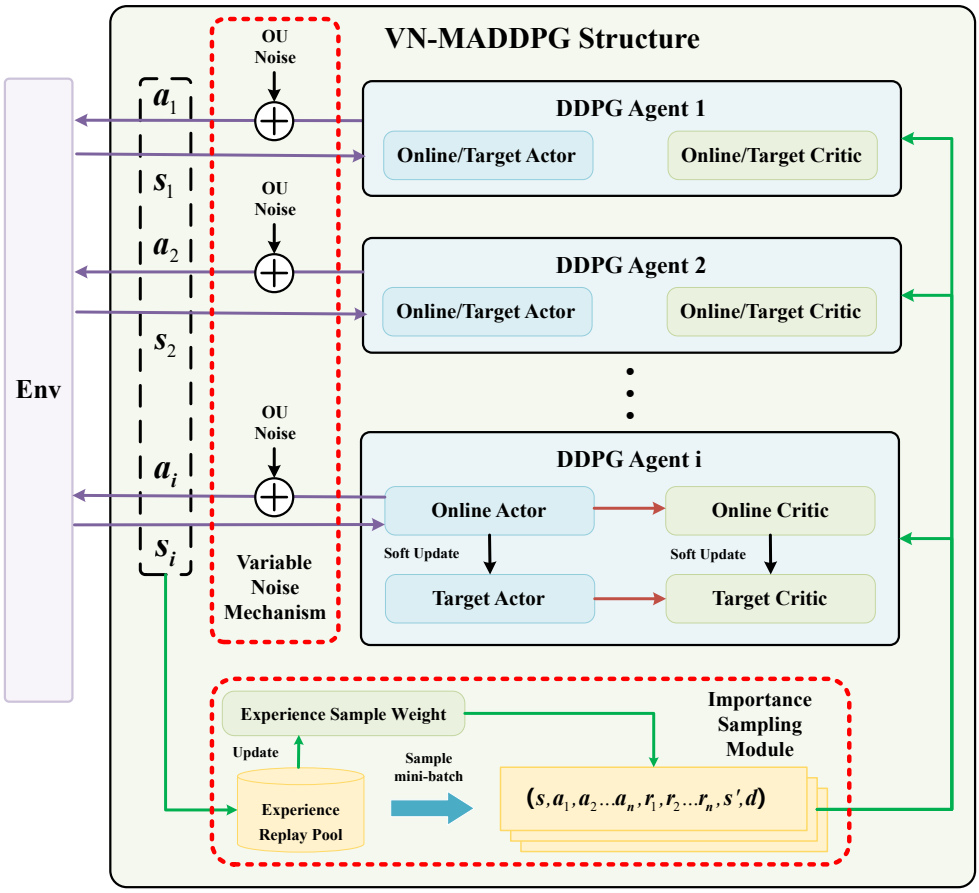

Figure 1. The structure and data flow of VN-MADDPG. We designed a variable-noise module that gradually reduces noise based on the proportion of remaining training rounds. We incorporated an importance sampling module into the experience replay buffer to focus more on samples that have a greater impact on the learning process.

3.1.1. Variable-Noise Mechanism

In reinforcement learning, the exploration-exploitation dilemma is a well-known challenge. It refers to the trade-off between exploring new actions to gather more information about the environment and exploiting the current best-known action to maximize rewards. Striking the right balance is crucial, especially in complex multi-agent scenarios where the dynamics are more unstable.

Effective exploration is critical for discovering optimal policies, especially in complex multi-agent environments. MADDPG utilizes fixed noise to encourage agents to explore the environment, increasing the randomness of their behavior. However, this static approach often leads to suboptimal performance because it does not adapt to the evolving needs of the learning process. The early stages of training require higher exploration to map out the environment, while the later stages benefit from reduced exploration to fine-tune the learned policy. A variable-noise adjustment mechanism that adapts the noise level based on the training progress is theoretically more sound.

In VN-MADDPG, we modify the noise to be dynamic. Before the start of each training episode, the agent retrieves the current episode number ep_i and the total number of training episodes $explor\_eps$ to calculate the noise value for the current round, as shown in Figure 2. As the number of training episodes increases, the noise decreases accordingly. The noise is calculated based on the initial noise, the final noise, and the remaining training episodes, as shown in the following equation:

$$Explr_{rem} = \frac{\max(0, explor\_eps - ep\_i)}{explor\_eps} \times 100\% \quad (1)$$

$$Noise = noise_{fnl} + (noise_{init} - noise_{fnl}) \times Explr_{rem} \quad (2)$$

$Explr_{rem}$ is the percentage of remaining training episodes. $explor\_eps$ is the total number of training episodes. ep_i is the number of current episodes. The $Noise$ of the current episode is equal to the final noise $noise_{fnl}$ plus the difference between the initial noise $noise_{init}$ and $noise_{fnl}$, multiplied by $Explr_{rem}$.

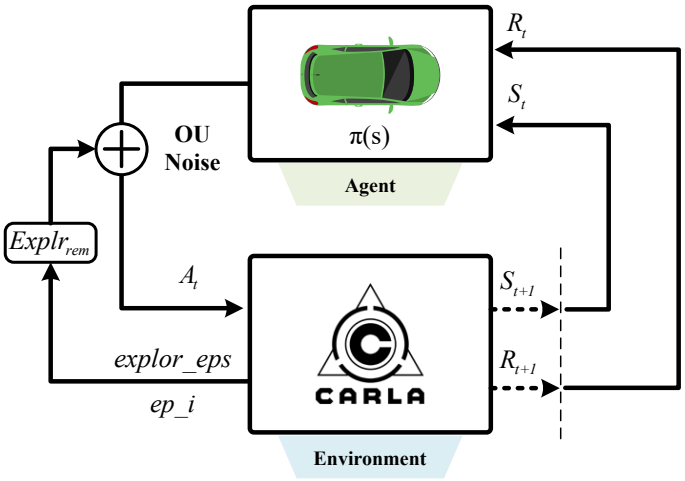

Figure 2. Variable-noise mechanism. For each episode, the agent vehicle calculates the OU noise value for the current episode based on the number of current episodes.

As the number of training episodes increases, the noise gradually diminishes to zero. This mechanism encourages the agent to explore the environment extensively at the start. In the early stages, larger noise allows the agent to try various actions, accumulating diverse experiences.

As training progresses, the agent continuously learns and optimizes its decision-making process through the policy network, gradually forming an effective strategy. In the later stages of training, with a more mature decision strategy in place, the noise decreases. This shift allows the agent to rely more on its learned policy rather than random exploratory actions.

The reduction in noise helps the agent perform tasks more reliably and with less unnecessary randomness, enhancing decision-making stability. Initially, the agent explores broadly to gather comprehensive environmental information. Later, it effectively uses the learned strategy to optimize decisions. This approach improves training efficiency and enhances the agent's performance in complex unsignalized intersections.

3.1.2. Importance Sampling Module

We also designed an experience importance sampling module to enhance the learning efficiency and convergence speed of the decision-making model. This module focuses on selecting experiences based on their sampling probability, which is determined by their priority.

In our approach, experiences that have a significant impact on agent behavior or represent near-optimal solutions are given higher priority. The priority of an experience is determined by the Temporal Difference (TD) error or reward prediction error. Experiences with larger prediction errors are assigned higher priority because they indicate that the model's predictions for these experiences are less accurate, thus necessitating further learning.

The prediction error, which quantifies the difference between the predicted and actual values, is calculated using the following formula:

$$\delta_t = |r_t + \gamma \cdot V_{t+1} - Q(s_t, a_t)| \tag{3}$$

where r_t is the current reward obtained. γ is the discount factor, which represents the value of the next state calculated using the goal network, and is the value of performing the action in the current state. Here, a_t equals $\pi(s_t)$, which represents the action selected in the current state s_t. π is the policy that the current agent has learned, and s_t is the current state.

In addition,

$$V_{t+1} = Q'(s_{t+1}, \pi'(s_{t+1})) \tag{4}$$

where $Q^{'}$is the target Q network, $\pi^{'}$ is the target policy network, and s_{t+1} is the next state. The priority weights $piror_i$ for the experience are defined as follows:

$$piror_i = (\delta_t + \varepsilon)^{\alpha} \tag{5}$$

where ε is a very small constant to ensure that the priority is not zero. α is a hyperparameter that controls the priority weights. It prevents high-priority experiences from being oversampled, which could result in training bias.

The probability of each experience being selected is calculated based on its priority weight $piror_i$, defined as follows:

$$P_i = \frac{piror_i}{\sum_{j=0}^{n} piror_j} \tag{6}$$

It helps the decision model to focus more on impactful samples during training, effectively utilizing experiences in the replay buffer.

Additionally, the experience replay buffer continuously updates the experience samples in the buffer based on their priority weights, as shown in Figure 3. This adaptive update mechanism prioritizes important experiences for learning, thereby enhancing overall learning efficiency, which helps VN-MADDPG achieve better strategies more quickly.

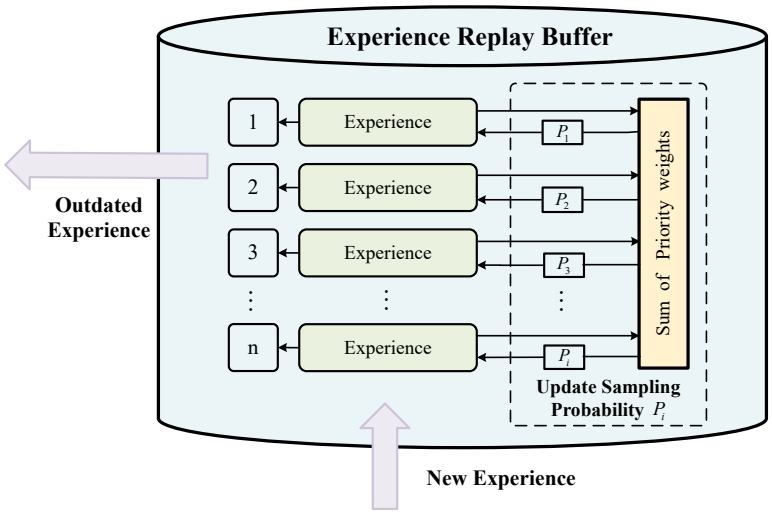

Figure 3. Updating the experience replay buffer. At each time step, experiences are stored in the replay buffer, and the sampling probabilities of all experiences are updated. When the replay buffer is full, outdated experiences are removed based on their importance weights.

The detailed steps of VN-MADDPG are summarized in Algorithm 1.

Algorithm 1 VN-MADDPG

Initialize all agents' actor network Q_{ω_i} and parameters ω_i
Initialize all agents' critic network Π_{θ_i} and parameters θ_i
Initialize all agents' target actor network $Q_{\omega_i'}$ and its parameters $\omega_i^{'}$
Initialize all agents' target critic network $\Pi_{\theta_i'}$ and its parameters $\theta_i^{'}$
Initialize the experience replay buffer D
for episode = 1 to N **do**
  Initialize the environment and obtain the initial observations
  for step = 1 to S **do**
    (1) Each agent gets $\mathbf{a_t}$ based on $\mathbf{s_t}$ and Q_{ω_i}
    (2) Execute $a_1, a_2, \ldots, a_N$, get $\mathbf{s}^{'}$, $r_1, \ldots, r_N$ and $\mathbf{d}$
    (3) Stored all to the replay buffer $\mathbf{D}$:
      $D \leftarrow (s, a_1, \ldots, a_N, r_1, \ldots, r_N, s^{'}, d)$
    (4) Update $piror_i$, $\mathbf{P_i}$ of experience
    if $experience\_num >= M$ **then**
      Sample from the D
      Update Π_{θ_i}:
        Calculate the target Q-value for each agent
        Calculate the loss L and update θ_i
      Update Q_{ω_i} by maximizing the expected r
      Soft-update target network:
        $\omega_i^{'} = \tau \cdot \omega_i + (1 - \tau) \cdot \omega_i^{'}$
        $\theta_i^{'} = \tau \cdot \theta_i + (1 - \tau) \cdot \theta_i^{'}$
    end if
  end for
end for

3.2. State Space

In an unsignalized intersection, multiple vehicles need to pass through the intersection at the same time. The state space for the vehicles in the scene needs to include the state data of all perceived vehicles. Taking any agent vehicle *ego* in the scene as an example, its state space is defined as follows:

$$\mathrm{S} = (V_{ego}, V_1, V_2 \ldots V_n, D_1, D_2 \ldots D_n, Dest_{ego}) \tag{7}$$

This includes the speed of the self-vehicle V_{ego}, the speeds of the other vehicles $V_1, V_2, \ldots, V_N$, the relative distances $D_1, D_2, \ldots, D_N$ of other vehicles from the self-vehicle, and the distance $Dest_{ego}$ of the destination from the self-vehicle.

3.3. Action Space

In reinforcement learning, the action space includes all possible actions that an agent can take. The agent's behavior is defined by the action space, and an accurate definition of the action space can facilitate the learning process. The agent can explore and exploit experiences more effectively to achieve its goals. The definition of the action space becomes more critical in multi-agent scenarios.

MADDPG is more suitable for solving continuous action-space problems in autonomous driving decision making. Its action space design is continuous. Taking any agent vehicle *ego* in the scene as an example, its action space is defined as follows:

$$\mathrm{A} = (Throttle_{ego}, Brake_{ego}, Steer_{ego}) \tag{8}$$

The action space for each agent vehicle in the scenario contains three continuous control signals: throttle $Throttle_{ego}$, brake $Brake_{ego}$, and steer $Steer_{ego}$. The agent makes acceleration, deceleration, and steering maneuvers based on these continuous control signals to cross the intersection and reach the destination smoothly to complete the task.

3.4. Reward Function

Based on the multi-agent task scenario, we define the following rewards.

Local reward: The task objective is for agent vehicles to successfully reach the destination from the starting point while passing through intersections safely and quickly. We incorporate vehicle speed and time spent as evaluation criteria. Agent vehicles receive a positive reward based on their proximity to the target point. The vehicle's proximity to the target point is used to determine whether it has reached the destination. If it does this successfully, it receives a reward for task completion. However, it incurs a penalty for conflicts with other vehicles.

Global reward: If all vehicles avoid collisions and safely reach their expected destination, each agent vehicle receives a positive reward. This promotes task cooperation among agent vehicles.

After multiple experimental adjustments, the final reward function is defined as follows:

$$\begin{aligned} r_i = r\_speed_i + r\_time_i - (distance_i - 1) \\ -r\_collision_i + r\_success_i + r\_success_{all} \end{aligned} \tag{9}$$

where the speed reward r_speed_i is a positive reward based on the difference between the agent's speed and the target speed; the time reward r_time_i is a positive reward based on the time taken by the agent to arrive at the destination safely; $distance_i$ rewards or penalizes based on the distance between the agent vehicle and the target point—if the distance is greater than 1 m, a penalty is given, and if it is less than 1 m, a positive reward is given, encouraging the agent vehicles to continue moving toward the goal after crossing the intersection; $r\_collision_i$ is a penalty incurred if the vehicle collides; $r\_success_i$ is a positive award given if the vehicle successfully reaches the target and completes the task; and $r\_success_{all}$ is a positive reward given if all agent vehicles successfully complete the task and arrive at their targets.

4. Experiments

In this section, we introduce the simulation environment and then we describe its rules in detail. Finally, we introduce our evaluation metrics.

4.1. Scenario Design

We evaluated the differences between various algorithms in a typical unsignalized two-way single-lane intersection with four entrances and four exits, as shown in Figure 4a.

The scenario includes three agent vehicles equipped with the algorithms. These vehicles can interact with each other. The decision-making algorithms' task is to adjust the longitudinal and lateral control of each vehicle along its driving path to ensure that each vehicle's actions at each step are aligned with the global optimal solution. The goal is for all vehicles to avoid collisions, pass through the intersection quickly, and reach their respective destinations.

To validate the effectiveness and superiority of VN-MADDPG, we utilized Python as the development language and constructed the algorithm network structure based on PyTorch. We chose the Town04 map in the CARLA simulation [38] and deployed different decision-making algorithms in an unsignalized two-way single-lane intersection. The compared decision-making algorithms include the DDPG algorithm, the MADDPG algorithm, and our proposed VN-MADDPG algorithm.

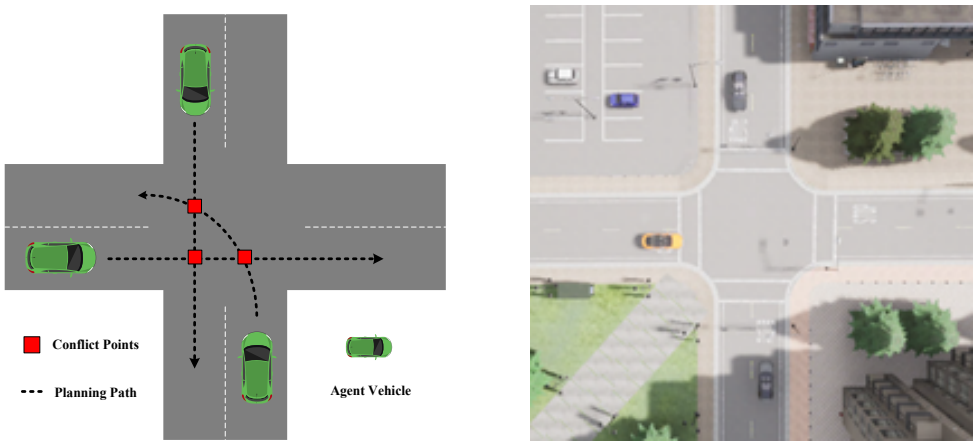

Figure 4. Unsignalized two-way single-lane intersection scenario. (**a**) Scenario design. There are three agent vehicles at this intersection. They have their own destinations and expected trajectories. The red dots in the figure are potential conflict points. (**b**) CARLA simulation. Experimental scenario of unsignalized two-way single-lane intersection in Town04.

We chose CARLA for its highly realistic urban traffic simulation, which includes diverse road types, intersections, and sensors, enabling comprehensive testing of our algorithm. Its compatibility with deep learning frameworks also facilitates effective data collection and model training. The effects of the simulation experiment scene are shown in Figure 4b.

The settings for the environment in the simulation were as follows:

- To ensure experimental efficacy, the destinations of each agent vehicle were fixed. An agent vehicle makes a left turn through the intersection, while the other two vehicles continue straight through the intersection. The agent vehicles need to pass through the intersection and reach their specified destinations while controlling both lateral and longitudinal movements to avoid collisions. The experimental setup aimed to create conflict points in vehicle trajectories as much as possible, covering various conflict scenarios that could occur at intersections.
- The initial lane of each agent vehicle was predetermined. At the beginning of each training episode, all vehicles were randomly generated within 5 m of the intersection on their respective lanes. This setup made the intersection more random and uncertain. The initial speed of all vehicles was around 3 m/s, making their speed and time when entering the intersection uncertain. The target speed for vehicles was set at 5 m/s, with a maximum speed limit of 8 m/s.
- Each agent vehicle in the environment was equipped with visual sensors and collision detection sensors, providing all the perception data required for the experiment. To ensure stable experimental results, only three agent vehicles were set up in the environment. This setup simplified the training scenario and ensured the model could learn more effectively.
- The main goal of the experiment was to compare the performance of various decision-making algorithms in identical experimental settings. The comparison focused on the collision and success rates of vehicles at intersections when deploying the different decision-making algorithms.

4.2. Evaluation Indicator

We utilized several metrics to assess the performance of autonomous vehicle decision-making algorithms in this scenario, including the collision rate, success rate, pass time, and cumulative reward.

The collision rate is defined as the proportion of collisions per 100 training episodes. Similarly, the success rate is defined as the proportion of the number all agents who complete the task per 100 training episodes. Task completion is defined as all vehicles successfully passing through the intersection and reaching their destinations without any collisions.

5. Experimental Results and Analysis

In this section, we analyze the results of the experiments. We first introduce the types of recorded data and the training parameters of the models. Then, we compare the training processes of the models in terms of the cumulative reward, collision rate, and success rate. Finally, we compare the decision-making effectiveness of the trained models.

During the training process, we recorded the number of collisions leading to task failures, the number of successful task completions after all vehicles passed through the intersection, and the time taken to complete the task. We also recorded the cumulative rewards for each agent per round, as well as the total cumulative rewards for all vehicles.

The DDPG, MADDPG, and VN-MADDPG algorithms were trained, and relevant data were recorded in the same simulation environment settings. The network training parameters are shown in Table 1.

Table 1. Training parameters.

Parameter	Value	Explanation
Training episode	10,000	Total episodes used for training the algorithm
Update frequency	90	Frequency of soft updates for target network parameters
Initial noise	0.25	Initial noise value for the variable-noise mechanism
Final noise	0.0	Final noise value for the variable-noise mechanism
Learning rate	0.0005	Rate of updating model parameters during training
Discount factor	0.01	Weighting factor for future rewards
Batch size	256	Number of training samples used in each iteration
Max explore step	13000	Maximum steps for agent exploration in the environment

The parameters of reinforcement learning algorithms significantly impact a model's performance. The parameters mentioned in Table 1 were determined after numerous experiments involving continuous trials and adjustments. Insufficient training episodes may lead to inadequate learning and poor performance, while excessive episodes might cause overfitting. An appropriate update frequency ensures more stable learning. A low initial noise value might result in insufficient exploration, leading to suboptimal learning, whereas a high final noise value could impede convergence.

The total number of training episodes was 10,000. We saved the cumulative rewards of each agent vehicle and calculated the total reward value every 20 rounds. The success rate and collision rate were saved every 100 rounds.

5.1. Cumulative Reward

A comparison of the global cumulative rewards obtained by the different algorithms during the training process is shown in Figure 5. The horizontal axis represents the training episode, and the vertical axis represents the total cumulative reward. The figure shows that the DDPG and MADDPG algorithms exhibited similar growth trends in terms of the total cumulative rewards. Both algorithms eventually achieved relatively high total reward values but required longer training times.

In the same scenario and environmental settings, VN-MADDPG, which incorporates an importance sampling module and a variable-noise mechanism, exhibited higher efficiency in environmental exploration. The convergence efficiency of the decision-making model was considerably improved, and its overall performance surpassed that of DDPG and MADDPG.

The VN-MADDPG algorithm learned robust and stable decision-making strategies more quickly. The agent vehicles utilized environmental features more quickly and effectively after actively exploring the environment because of the importance sampling module and dynamically varying noise mechanism. After achieving a high total cumulative reward in training, the model remained very stable with minimal fluctuations in the reward values. The agent vehicles achieved higher reward values by collaborating with each other.

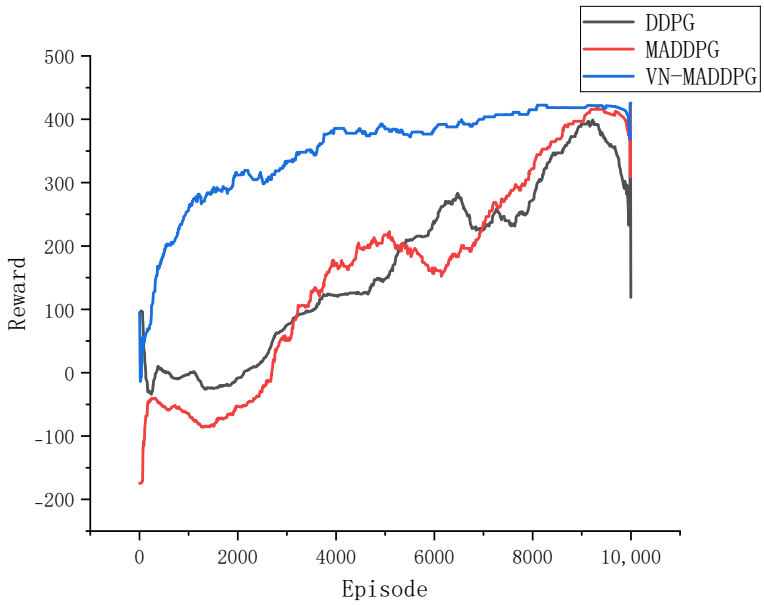

Figure 5. Comparison of total cumulative rewards of DDPG, MADDPG, and VN-MADDPG.

The average cumulative reward statistics obtained from the various decision-making algorithm models are shown in Table 2.

Table 2. Comparison of average cumulative rewards.

Algorithm	Average Value
DDPG	388.95
MADDPG	385.03
VN-MADDPG	421.09

Vehicles using the DDPG algorithm lacked cooperation, resulting in relatively poor decision-making performance. The learned policy of the agents tended to be self-centered, leading to a lower total cumulative reward. The decision-making performance of agent vehicles using the MADDPG algorithm was also not satisfactory. Although it promoted cooperation among vehicles, the resulting decision-making policy was not robust enough, and the training process was relatively slow.

5.2. Collision Rate

The VN-MADDPG algorithm demonstrated superior performance in reducing collision rates. As shown in Figure 6, its final collision rate was reduced to around 3%.

The cooperation between vehicles and the importance sampling of the replay buffer enabled efficient strategy iteration. Vehicles in the scenario avoided collisions with each other and reached their destinations quickly.

Compared to the DDPG and MADDPG algorithms, our algorithm substantially reduced the occurrence of collisions between vehicles in the early stages of training. It explored more robust decision-making strategies more quickly.

The ANOVA test revealed a significant difference in collision rates among the three algorithms. The F value of 36.41646 with a *p*-value of less than 0.0001 suggests that the improvements in the collision rates are statistically significant.

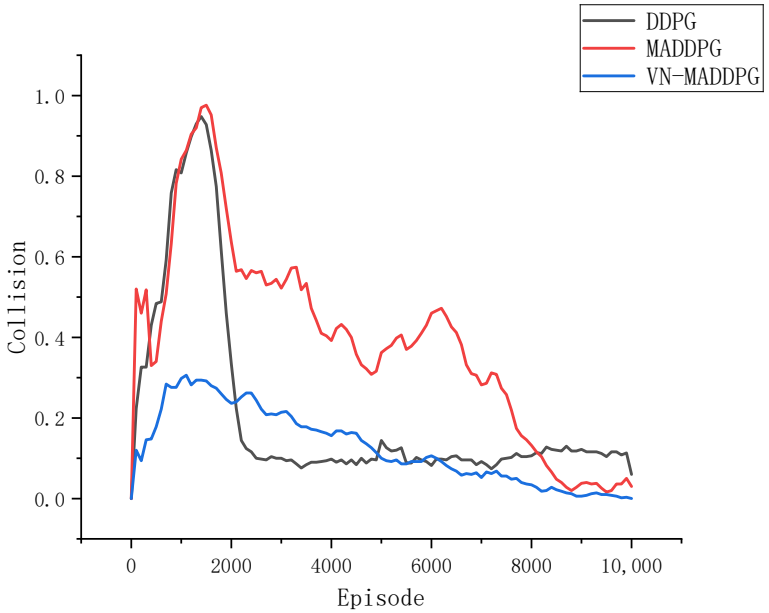

Figure 6. Comparison of collision rates of DDPG, MADDPG, and VN-MADDPG.

5.3. Success Rate

As shown in Figure 7, the trend of the changes in the success rate is similar to that in the cumulative reward. The DDPG and MADDPG algorithms exhibited a relatively slow convergence rate toward optimal policies. Their success rates in decision making fluctuated considerably, indicating instability. Additionally, the final policies derived from these algorithms were not particularly impressive.

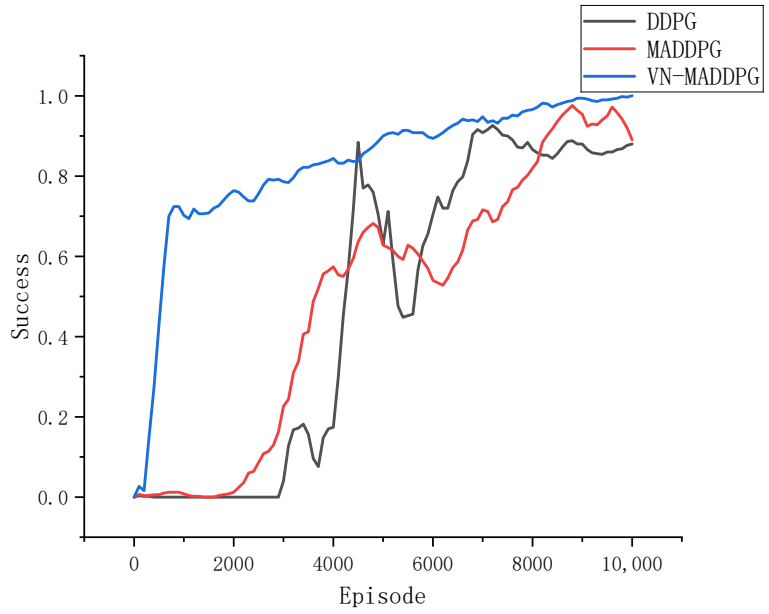

Figure 7. Comparison of traffic success rates of DDPG, MADDPG, and VN-MADDPG.

The VN-MADDPG algorithm focused more on experiences crucial for learning strategies. As the noise dynamically decreased, the VN-MADDPG algorithm relied more on its policy network to make final decisions. Our algorithm enabled agent vehicles in the

scenario to achieve a higher pass rate more quickly, maintaining a high success rate of around 97%.

The ANOVA test for success rates also demonstrated a significant difference among the algorithms. The F value of 37.6365 with a *p*-value of less than 0.0001 confirms that the improvements in success rates are statistically significant.

5.4. Validation of Trained Models

The trained models of the three decision-making algorithms were redeployed and tested in the same signal-free intersection scenario.

As shown in Table 3, our algorithm shows improvements in evaluation metrics such as the success rate, collision rate, and pass time.

Table 3. Comparison of collision rates, success rates, and pass times.

Algorithm	Collision Rate (%)	Success Rate (%)	Pass Time (s)
DDPG	7.2	92.8	1.2
MADDPG	9	91	1.27
VN-MADDPG	3	97	1.19

The VN-MADDPG algorithm significantly enhances the utilization of experience samples in multi-agent deep reinforcement learning algorithms, strengthening the ability of multi-agent deep reinforcement learning algorithms to cope with dynamically changing scenarios. VN-MADDPG enhances the learning speed and convergence efficiency of decision-making models for autonomous vehicles. The decision-making strategies learned by agent vehicles are more robust.

6. Conclusions

We propose the VN-MADDPG algorithm to address the challenges of instability and suboptimal decision making for autonomous vehicles at unsignalized intersections. Based on the MADDPG framework, VN-MADDPG includes a variable-noise mechanism and an importance sampling module to enhance stability and robustness. The variable-noise mechanism dynamically adjusts the level of exploration based on the training progress, promoting extensive exploration in the early stages and gradually shifting to reliance on learned strategies as training advances. The importance sampling module prioritizes impactful experiences, thereby improving learning efficiency and accelerating convergence. These enhancements collectively contribute to more reliable and effective decision making in complex and dynamic intersections.

We deployed VN-MADDPG in the CARLA simulation platform. We verified the effectiveness and superiority of our method by comparing it with the DDPG and MADDPG algorithms. Experimental results demonstrate that the VN-MADDPG algorithm effectively enhances the decision-making ability of autonomous vehicles at unsignalized intersections. The decision-making algorithm enables better adaptation to dynamic environments and unexpected situations. It improves the success rate and efficiency of autonomous vehicles passing through intersections.

In the future, we may consider factors such as reduced wheel–road adhesion in the simulation platform to more realistically simulate real-world conditions [39]. Exploring improvements in braking and turning stability under these conditions could further enhance the effectiveness of decision-making algorithms. This could be a promising research direction.

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

Funding: This research was funded by the Vehicle–Road Cooperative Autonomous Driving Fusion Control Project.

Data Availability Statement: The CARLA simulator used can be obtained from https://github.com/carla-simulator/carla, accessed on 7 August 2024. The code for our experimental project and the VN-MADDPG algorithm are available on GitHub at https://github.com/l-ZhangHao-l/VN-MADDPG, accessed on 7 August 2024.

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

References

1. Chen, S.; Hu, X.; Zhao, J.; Wang, R.; Qiao, M. A review of decision-making and planning for autonomous vehicles in intersection environments. *World Electr. Veh. J.* **2024**, *15*, 99. [CrossRef]
2. Wei, L.; Li, Z.; Gong, J.; Gong, C.; Li, J. Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 44–51.
3. Kala, R. *On-Road Intelligent Vehicles: Motion Planning for Intelligent Transportation Systems*; Butterworth-Heinemann: Oxford, UK, 2016.
4. Chen, L.; Englund, C. Cooperative intersection management: A survey. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 570–586. [CrossRef]
5. Administration, N. Fatality Analysis Reporting System. [Online]. 2018. Available online: https://www-fars.nhtsa.dot.gov/ (accessed on 7 August 2024).
6. He, J.-Y.; Cheng, Z.-Q.; Li, C.; Xiang, W.; Chen, B.; Luo, B.; Geng, Y.; Xie, X. Damo-streamnet: Optimizing streaming perception in autonomous driving. *arXiv* **2023**, arXiv:2303.17144.
7. Li, C.; Cheng, Z.-Q.; He, J.-Y.; Li, P.; Luo, B.; Chen, H.; Geng, Y.; Lan, J.-P.; Xie, X. Longshortnet: Exploring temporal and semantic features fusion in streaming perception. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
8. Lv, H.; Du, Y.; Ma, Y.; Yuan, Y. Object detection and monocular stable distance estimation for road environments: A fusion architecture using yolo-redeca and abnormal jumping change filter. *Electronics* **2024**, *13*, 3058. [CrossRef]
9. Li, N.; Yao, Y.; Kolmanovsky, I.; Atkins, E.; Girard, A.R. Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1428–1442. [CrossRef]
10. Kerner, B.S. Failure of classical traffic flow theories: Stochastic highway capacity and automatic driving. *Phys. A Stat. Mech. Its Appl.* **2016**, *450*, 700–747. [CrossRef]
11. Mo, C.; Li, Y.; Zheng, L. Simulation and analysis on overtaking safety assistance system based on vehicle-to-vehicle communication. *Automot. Innov.* **2018**, *1*, 158–166. [CrossRef]
12. Xue, Y.; Zhang, X.; Cui, Z.; Yu, B.; Gao, K. A platoon-based cooperative optimal control for connected autonomous vehicles at highway on-ramps under heavy traffic. *Transp. Res. Part C Emerg. Technol.* **2023**, *150*, 104083. [CrossRef]
13. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Abbeel, O.P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
14. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
15. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
16. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
17. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
18. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 321–384.
19. Li, G.; Li, S.; Li, S.; Qu, X. Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient. *IET Intell. Transp. Syst.* **2022**, *16*, 1669–1681. [CrossRef]
20. Gutiérrez-Moreno, R.; Barea, R.; López-Guillén, E.; Araluce, J.; Bergasa, L.M. Reinforcement learning-based autonomous driving at intersections in carla simulator. *Sensors* **2022**, *22*, 8373. [CrossRef] [PubMed]
21. Xiao, W.; Yang, Y.; Mu, X.; Xie, Y.; Tang, X.; Cao, D.; Liu, T. Decision-making for autonomous vehicles in random task scenarios at unsignalized intersection using deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 7812–7825. [CrossRef]
22. Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; De Cote, E.M. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv* **2017**, arXiv:1707.09183.
23. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [CrossRef]
24. Yadav, P.; Mishra, A.; Kim, S. A comprehensive survey on multi-agent reinforcement learning for connected and automated vehicles. *Sensors* **2023**, *23*, 4710. [CrossRef]

25. Wang, J.; Zhang, Q.; Zhao, D. Highway lane change decision-making via attention-based deep reinforcement learning. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 567–569. [CrossRef]
26. Chen, D.; Hajidavalloo, M.R.; Li, Z.; Chen, K.; Wang, Y.; Jiang, L.; Wang, Y. Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11623–11638. [CrossRef]
27. Dai, Z.; Zhou, T.; Shao, K.; Mguni, D.H.; Wang, B.; Jianye, H. Socially-attentive policy optimization in multi-agent self-driving system. In Proceedings of the Conference on Robot Learning, Atlanta, GA, USA, 6–9 November 2023; pp. 946–955.
28. Toghi, B.; Valiente, R.; Sadigh, D.; Pedarsani, R.; Fallah, Y.P. Social coordination and altruism in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24791–24804. [CrossRef]
29. Guan, Y.; Ren, Y.; Li, S.E.; Sun, Q.; Luo, L.; Li, K. Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12597–12608. [CrossRef]
30. Antonio, G.-P.; Maria-Dolores, C. Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections. *IEEE Trans. Veh. Technol.* **2022**, *71*, 7033–7043. [CrossRef]
31. Zhuang, H.; Lei, C.; Chen, Y.; Tan, X. Cooperative decision-making for mixed traffic at an unsignalized intersection based on multi-agent reinforcement learning. *Appl. Sci.* **2023**, *13*, 5018. [CrossRef]
32. Hu, J.; Hu, S.; Liao, S.-W. Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods. *arXiv* **2021**, arXiv:2106.14334.
33. Wu, T.; Jiang, M.; Zhang, L. Cooperative multiagent deep deterministic policy gradient (comaddpg) for intelligent connected transportation with unsignalized intersection. *Math. Probl. Eng.* **2020**, *2020*, 1820527. [CrossRef]
34. Hu, W.; Mu, H.; Chen, Y.; Liu, Y.; Li, X. Modeling interactions of autonomous/manual vehicles and pedestrians with a multi-agent deep deterministic policy gradient. *Sustainability* **2023**, *15*, 6156. [CrossRef]
35. Liu, J.; Hang, P.; Na, X.; Huang, C.; Sun, J. Cooperative decision-making for cavs at unsignalized intersections: A marl approach with attention and hierarchical game priors. *Authorea Prepr.* **2023**. [CrossRef]
36. Orr, J.; Dutta, A. Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors* **2023**, *23*, 3625. [CrossRef]
37. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
38. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. Carla: An open urban driving simulator. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.
39. Pugi, L.; Favilli, T.; Berzi, L.; Locorotondo, E.; Pierini, M. Brake blending and torque vectoring of road electric vehicles: A flexible approach based on smart torque allocation. *Int. J. Electr. Hybrid Veh.* **2020**, *12*, 87–115. [CrossRef]