Decision-Making for Complex Scenario using Safe Reinforcement Learning

Jie Xu
School of Autonomous Engineering
Wuhan University of Techonology
Wuhan, China
xj305305@163.com

Xiaofei Pei*
School of Autonomous Engineering
Wuhan University of Techonology
Wuhan, China
peixiaofei7@whut.edu.cn

Kexuan Lv
School of Autonomous Engineering
Wuhan University of Techonology
Wuhan, China
2792328835@qq.com

2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI) | 978-1-7281-8497-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/CVCI51460.2020.9338584

***Abstract*—In recent years, machine learning is widely used in many fields. Compared with the rule-based method, machine learning plays a more excellent role in the decision-making of the autonomous vehicle. Some complex situations are often met in our daily life. To this end, Safe reinforcement learning(RL) is introduced to ensure that safer actions are selected. Constant Turn Rate and Acceleration(CTRA) model is first used to predict the future trajectories of surrounding vehicles. Then Double Deep Q-Learning(DDQN) method is used to make decisions and ensure the autonomous vehicle can move at the desired speed as much as possible. In order to achieve a safer decision-making, some safety rules are introduced. Finally, the algorithm is demonstrated in Simulation of Urban Mobility(SUMO) and has been proved to have an outstanding performance on such a complex scenario.**

Keywords—decision-making, the autonomous vehicle, safe reinforcement learning, CTRA model.

I. INTRODUCTION

Decision-making, as a key part of autonomous driving technology, is particularly important to improve driving safety and reduce the time required. Compared with traditional rule-based methods, reinforcement learning(RL) has a better performance in decision-making. How to use RL to make better decisions in a complex scenario is the focus of this paper.

RL has great advantages in solving discrete decision-making problems. Double Deep Q-Learning(DDQN) method, as one of the most important methods in RL for solving discrete decision-making problems, has been used by more and more researchers. [1] used DDQN method to make the autonomous vehicle learn to make decisions in the direct interaction with the simulation environment. [2] proposed an exploration strategy for discretionary lane change based on DDQN method and had a relatively good performance. A driving policy based on DDQN method was proposed in [3] for a mixed driving environment where the road is occupied by the autonomous vehicles and the manual vehicles randomly.

Although RL can provide an effective strategy, it fails to guarantee that the selected action at every moment is safe. For this, a probabilistic model checker was proposed in [4] to calculate the probability that action selected can safely achieve the goal and then choose the action with the largest probability value. [5] defined safety restrictions according to the Gipps model which makes the autonomous vehicle keep a safe distance from the vehicle in front. [6] proposed two action-value functions which are used to maximize the cumulative discount positive reward and the cumulative discount negative reward to achieve a safer action. In [7], the autonomous vehicle needed to maintain a safe distance from other vehicles to ensure that the action selected by the agent at any time is safe.

In this paper, safe RL is used to make decisions in a seven-vehicle scenario where three lanes are reduced to two lanes. The purpose is to ensure that the autonomous vehicle can drive safely at the desired speed as much as possible. The main contributions of this paper are: 1) The future trajectory information of surrounding vehicles will be achieved by the agent in time. 2) A more comprehensive reward function is formulated to decide whether the action selected is good. 3) Some safety rules are used to improve the security of the agent's actions at every moment.

The rest of this paper is organized as follows. Section II gives a brief introduction of RL and the strategies used. Section III lists the problem formulation and safety rules. Section IV is the analysis and comparison of training results. Section V gives the summary of this paper.

II. BACKGROUND

A. Markov Decision Process

RL is used to solve the problem of decision-making [8]. Its basic model is a Markov decision process(MDP), which can be defined by a tuple$<S, A, P, R, \gamma>$, where S represents a state space, A represents an action space, $P(s_{t+1}| s_t, a_t)$ represents a state transition model, R represents a reward function, γ represents a discount factor and its value is between 0 and 1 [9].

The purpose of RL is to learn a strategy π to ensure the maximum value of the reward function accumulated in the long run [10]. The action-value function which yields the policy π can be defined as:

$$Q_\pi(s,a) = E_\pi\left\{\sum_0^\infty \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\} \tag{1}$$

The optimal action-value function which yields the optimal policy π^* can be defined as:

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \tag{2}$$

B. DDQN

Deep Q-learning(DQN) is a neural network that outputs an action-value $Q(s, \cdot; \theta)$ according to a given state s, where θ represents the neural network parameters. However, action selection and action evaluation in DQN are both based on the target value network parameter, it will lead to overestimation.

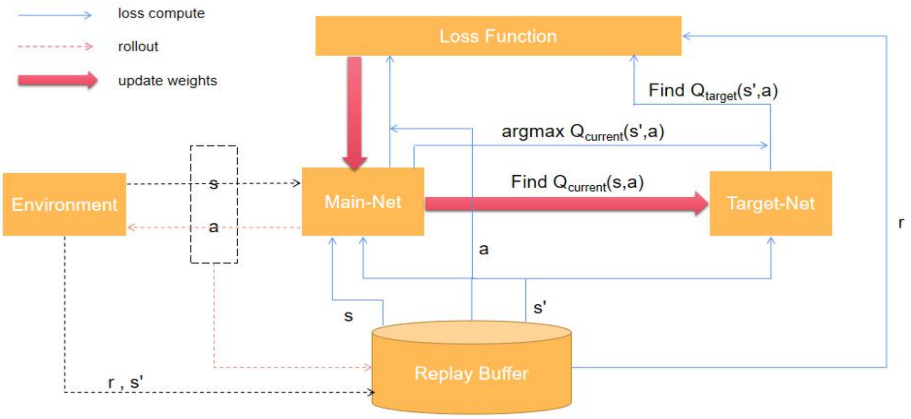


Fig. 1. The framework of DDQN

To this end, DDQN method is proposed and shown in Fig. 1 which has the same two Q-network architectures as DQN. It decomposes the calculation method of Q-value into two parts. In the DDQN, the main network parameter θ is used to select the action and the target network parameter θ^{-} is used to evaluate the target TD-error [11]. The target TD-error used in DDQN is defined as:

$$y_t^{DDQN} = r + \gamma \max_{a'} \overset{\Lambda}{Q}(s', \arg\max_{a\in A} Q(s',a \mid \theta^{-}) \mid \theta^{-}) \quad (3)$$

In this paper, prioritized experience replay is used to judge which experience has greater contributions to the learning process according to the value of TD-error. Experience with high learning efficiency will be given greater sampling weight to improve the utilization efficiency of empirical data [12].

III. PROBLEM FORMULATION

A. System Architecture

The system architecture of safe RL is shown in Fig. 2. The environment provides the state which contains two parts, one is the current state of the autonomous vehicle and the surrounding vehicles, the other is the future state of the surrounding vehicles which is predicted by Constant Turn Rate and Acceleration(CTRA) model to the agent. Then the agent will select an action according to these state information. When the action is decided by the agent, it will be checked by the safety rules. If the action fails to meet the safety rules, the safety rules will replace the selected action with a safer action. Otherwise, the autonomous vehicle will drive with the action selected by the agent. After the end of the step, the agent will receive the reward about the selected action.

B. Scenario

In this paper, the decision-making of the autonomous vehicles is used in the scenario shown in Fig. 3. The scenario is built in Simulation of Urban Mobility(SUMO). The initial speed of the autonomous vehicle is set to 15(m/s) and the initial speeds from Vehicle1 to Vehicle6 are all set between [12(m/s), 14(m/s)].

C. State

For the agent, if it knows the future trajectories of surrounding vehicles, it will choose a better action. In this paper, CTRA model is used to predict surrounding vehicles' future trajectories of next 1s. Under the premise that the speed and yaw rate of the vehicle are not affected, the position of the vehicle at each time step can be predicted based on the vehicle kinematics model. The longitudinal displacement of the vehicle is defined as:

$$\begin{aligned}\Delta x(t) = \frac{1}{\omega^2}[(v\omega + \alpha\omega t)\sin(\theta + \omega t) + \\ \alpha\cos(\theta + \omega t) - v\omega\sin\theta - \alpha\cos\theta]\end{aligned} \quad (4)$$

where ω represents the yaw rate of the vehicle, v represents the speed of the vehicle, a represents the acceleration of the vehicle, t represents the time step, θ represents the heading angle of the vehicle

State space S is defined as:

$$S = [x, v, l, x_i, x_{i,t}, v_i], i = 1,2,...,6, t = 1,2,...,10 \quad (5)$$

where x represents the longitudinal position of the autonomous vehicle, v represents the speed of the autonomous vehicle, l represents the lane of the autonomous vehicle, x_i and v_i respectively represent the longitudinal position and speed of the surrounding vehicle at six positions of the autonomous vehicle: left-front, front, right-front, left-behind, behind and right-behind, $x_{i,t}$ represents the longitudinal position of the vehicle at the corresponding position of i after t time steps in the future(each time step is set to 0.1s). If vehicle at a certain position does not exist, x_i and $x_{i,t}$ is set to 999(m) and v_i is set to 0(m/s).

D. Action

Action space A is defined as:

$$A = [a_i, y_{lc}], i = 1,2,3,4,5 \quad (6)$$

where action space includes two parts, a_i represents speed change and y_{lc} represents lane change. For speed change, five kinds of accelerations are given to choose: -2m/s^2, -1m/s^2, 0m/s^2, 1m/s^2 and 2m/s^2. For lane change, three options are

978-1-7281-8497-5/20/$31.00 © 2020 IEEE

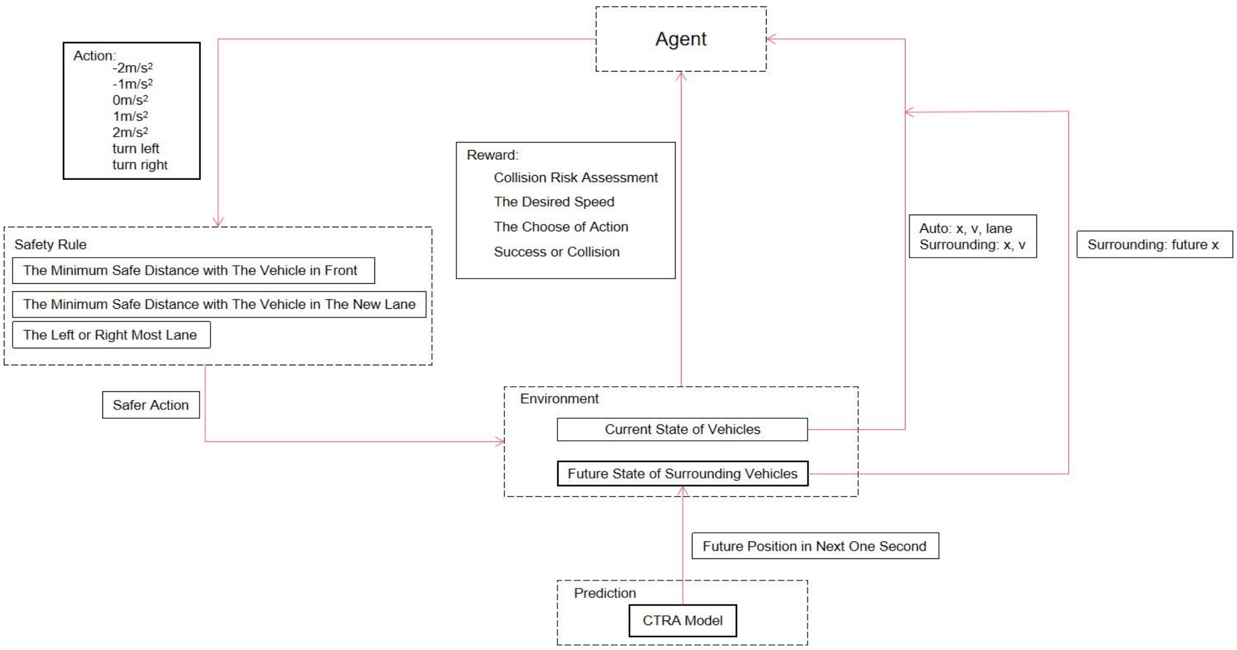
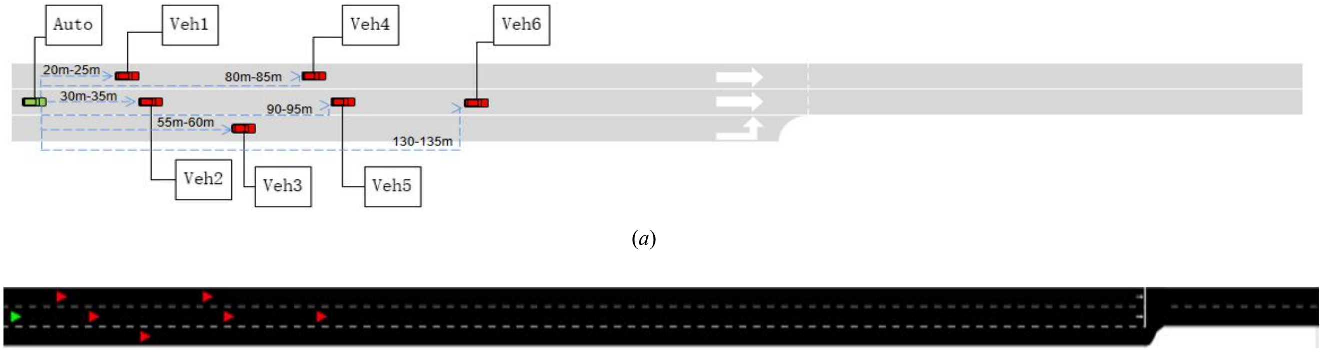

Agent
Action:
-2m/s²
-1m/s²
0m/s²
1m/s²
2m/s²
turn left
turn right
Reward:
Collision Risk Assessment
The Desired Speed
The Choose of Action
Success or Collision
Safety Rule
The Minimum Safe Distance with The Vehicle in Front
The Minimum Safe Distance with The Vehicle in The New Lane
The Left or Right Most Lane
Auto: x, v, lane
Surrounding: x, v
Surrounding: future x
Safer Action
Environment
Current State of Vehicles
Future State of Surrounding Vehicles
Future Position in Next One Second
Prediction
CTRA Model

Fig. 2. The system architecture of safe reinforcement learning



(*a*)

(*b*)

Fig. 3. A complex scenario where green represents the autonomous vehicles and red represents manually vehicles

(a)Schematic diagram of simulation scenario, (b) Simulation scenario built in SUMO

given: change lane to the left, keep the lane and change lane to the right.

E. Reward Function

The reward function plays an important role in whether the agent can reach the goal, so some reasonable reward functions needed to be set. In this paper, the autonomous vehicle is hoped to drive at the desired speed without collision. In response to this, the reward function is set from the following aspects:

1) Collision risk assessment: For a more comprehensive collision risk assessment, three safety factors are used. The first safety factor is time to collision(TTC), which is defined as follows: the time required to collide if two vehicles continue to drive at the current speed and lane [13]. The formula of TTC is defined as:

$$TTC = \frac{\Delta D_{nearest}}{v_{ego} - v_{nearest}} \qquad (7)$$

where $\triangle D_{nearest}$ represents the relative distance of the nearest vehicle in front, v_{ego} represents the speed of the autonomous vehicle, and $v_{nearest}$ represents the speed of the nearest vehicle in front.

The second safety factor is minimal safe distance(MSD), which is defined as follows: the minimal safe distance between two vehicles.

The third safety factor is inter vehicular time(IVT), which is defined as follows: the time required to collide if the vehicle in front stops and the vehicle in behind continues to drive at the current speed and lane. The formula of IVT is defined as:

$$IVT = \frac{\Delta D_{nearest}}{v_{ego}} \qquad (8)$$

The safety factors of MSD and IVT are supplements to TTC. For example, if there is a traffic accident in front, two vehicles still need to maintain a minimal safe distance. In addition, when two vehicles both move at a same fast speed and are very close, although the value of TTC is large, it is still very dangerous.

For the above three safety factors, risk values of each safety factor are defined as :

$$R_{TTC}=\begin{cases}1 & if \quad TTC<1.5s\\ 0 & if \quad otherwise\end{cases} \tag{9}$$

$$R_{MSD}=\begin{cases}1 & if \quad MSD<10m\\ 0 & if \quad otherwise\end{cases} \tag{10}$$

$$R_{IVT}=\begin{cases}1 & if \quad IVT<0.6s\\ 0 & if \quad otherwise\end{cases} \tag{11}$$

The collision risk can be divided into a forward collision risk and a backward collision risk. For the backward collision risk, the safety factor IVT has no effect. Therefore the corresponding reward function is set as follows:

$$R_F=\begin{cases}-1 & if \quad R_{TTC}+R_{MSD}+R_{IVT}\geq 1\\ 0 & if \quad otherwise\end{cases} \tag{12}$$

$$R_B=\begin{cases}-1 & if \quad R_{TTC}+R_{MSD}\geq 1\\ 0 & if \quad otherwise\end{cases} \tag{13}$$

*2) The desired speed: T*he autonomous vehicle is hoped to be able to drive at the desired speed as much as possible. The corresponding reward function is set as follows:

$$R_v=-2\times\left|v_{ego}-v_{desire}\right| \tag{14}$$

where v_{ego} represents the current speed of the autonomous vehicle, and v_{desire} represents the desired speed of the autonomous vehicle. According to the normal vehicle speed, 21(m/s) is the desired speed here.

3) The choose of action: The autonomous vehicle is hoped to accelerate to the desired speed and drives at the desired speed at much as possible while avoiding meaningless lane changes. The corresponding reward function is set as follows:

$$R_a=\begin{cases}-a^2 & if \quad a=-2m/s^2 \ or \ a=-1m/s^2\\ a^2 & if \quad a=2m/s^2 \ or \ a=1m/s^2\\ -0.5 & if \quad lane\ change\\ 0 & if \quad otherwise\end{cases} \tag{15}$$

where a represents the acceleration of the autonomous vehicle.

4) Success or collision: When the autonomous vehicle collides with other vehicles or fails to change the lane when the vehicle lane is reduced, both cases will be regarded as collision, and then a large reward function will be s as follows:

$$R_r=-300 \tag{16}$$

The final reward function corresponding to time step t is set as follows:

$$R=\omega_1 R_F+\omega_2 R_B+\omega_3 R_v+\omega_4 R_a+\omega_5 R_{\mathrm{r}} \tag{17}$$

In this paper, the weights are determined through trial and error process as follows: ω_1=0.4, ω_2=0.4, ω_3=0.13, ω_4=0.1, ω_5=1.

F. Safety Rule

When the agent chooses an action, it may cause the autonomous vehicle to collide with other vehicles or leave the lane. In order to ensure that the agent can choose a safer action, the following safety rules are proposed:

1) The minimum safe distance with the vehicle in front: When the speed of the autonomous vehicle is faster than the speed of the vehicle in front and violates the minimum safe distance, it is very dangerous. In order to avoid a collision at some time later, the inequality should be satisfied as follows:

$$v_{ego}\times t-v_{front}\times t+\frac{1}{2}a_{\max}\times t^2>0 \tag{18}$$

where v_{ego} represents the speed of the autonomous vehicle, v_{front} represents the speed of the vehicle in front which is in the same lane, t represents the time interval and $a_{\max}$ represents the maximum deceleration of the autonomous vehicle. The minimum safe time interval $t_{\min}$ should be satisfied as follows:

$$t_{\min}=\inf\left\{t:t>\frac{2(v_{ego}-v_{front})}{a_{\max}}\right\} \tag{19}$$

The corresponding minimum safe distance $d_{\min}$ should be satisfied as follows:

$$d_{\min}=(v_{ego}-v_{front})\times t_{\min} \tag{20}$$

When the relative distance between the autonomous vehicle and the vehicle in front which is in the same lane is less than the minimum safe distance, the autonomous vehicle will drive at the maximum deceleration, otherwise it will drive with the action selected by the agent.

2) The minimum safe distance with the vehicle in the new lane: When the autonomous vehicle chooses to change the lane, it needs to consider whether it will collide with vehicles in front or behind in the new lane. In order to avoid this situation, it is important to determine whether the autonomous vehicle meets the minimum safety distance with the vehicles in front or behind after it changing the lane. If it fails to meet the minimum safety distance, the autonomous vehicle will cancel the lane change and continue to drive at the current lane with the same speed.

Otherwise, the autonomous vehicle will drive with the action selected by the agent.

3) In the left or right most lane: When the autonomous vehicle is in left most lane, if the agent chooses to change lane to the left, it will let the autonomous vehicle deviate from the lane. The agent is set to cancel the lane change and continue to drive at the current lane with the same speed. This solution also applies to the right most lane.

IV. EVALUATION

In this paper, our strategy is evaluated in the scenario designed in Section III. Python is used to write programs and build the scenario in SUMO. The system runs on a Inter Xeon E5-1650 v4 CPU with 3.6Ghz. The simulation time of each step is set to 30s and the duration of one-step decision is set to 0.1s. The DDQN model totally has three layers. The cells of the first dense layer is 256, the cells of the second dense layer is 128 and the final dense output layer contains as many cells as the number of the action size. A group of relatively good hyper-parameter values was selected after many experiments and Table I summarizes them.

Fig. 4 shows the training curve of average cumulative reward of CTRA+safe RL. A total of 8000 episodes have been carried out in this paper. It can be seen that training result has less fluctuations after 6500 episodes.

The models of traditional RL, safe RL, safe RL with limited perception(The perceivable distance of the autonomous vehicle is set to 40m), CTRA+safe RL and CTRA+safe RL with more actions(This model is used to simulate the driving behavior of aggressive drivers and give nine kinds of accelerations to choose: -4m/s^2, -3m/s^2, -2m/s^2, -1m/s^2, 0m/s^2, 1m/s^2, 2m/s^2, 3m/s^2 and 4m/s^2) are compared. The trained models are tested 500 episodes in the scenario built in SUMO. Free collision rate, average speed and average speed variance in 500 episodes are comparison indicators in this paper and the results are shown in TABEL II.

TABLE I. HYPER-PARAMETER VALUES

Hyper-parameter	**Value**
activation function	Adam
minibatch size	32
target network soft update	0.01
discounter factor	0.995
learning rate	0.0001
exploration	0.5→0.1(annealed over 2e5 steps)
prioritized replay Size	0.6

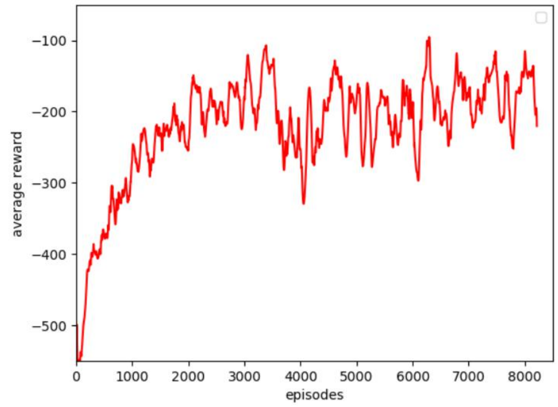


Fig. 4. The curve of CTRA+safe RL during training

TABLE II. TEST RESULTS OF TRAINED MODEL

	Free Collision rate(％)	**Average speed(m/s)**	**Average speed variance (m^2/s^2)**
Traditional RL	88	16.16	0.032
Safe RL with Limited Perception	91	18.82	0.025
Safe RL	97.8	19.56	0.070
CTRA+Safe RL	100	20.29	0.015
CTRA+Safe RL with More Actions	100	20.73	0.007

By comparing traditional RL and safe RL, it can be seen that the free collision rate ranges from 88％ to 97.8％ and the average speed increases from 16.16(m/s) to 19.56(m/s) which has a great improvement in safety reinforcement learning. It means the safety rules set can ensure that the selected action is relatively safe and play an important role in helping the autonomous vehicle pass the scenario at the desired speed as much as possible. When the perceivable distance of the autonomous vehicle is limited to 40m, not only the free collision rate reduces, but also the average speed becomes slower. Due to the limited perception, the vehicle fails to consider the unknown area, so the action selected will be more one-sided.

From the result of safe RL and CTRA+safe RL, knowing the future trajectories of surrounding vehicles plays an important role in the decision-making of the autonomous vehicle. When CTRA model is used to predict the trajectories of surrounding vehicles, the free collision rate ranges from 97.8 ％ to 100 ％ and the average speed increases 0.73(m/s) relative to safe RL. In this paper, a model of CTRA+safe RL with more actions is set. Although it has a faster average speed and a smoother velocity variance when more accelerations are given to choose, it will influence the comfortable of passengers and increase fuel consumption because of large acceleration or deceleration. Several sets of test results can suggest that CTRA+safe RL has a relatively good performance.

V. CONCLUSIONS

In this paper, a method based on safe RL is proposed for a complex scenario where the number of vehicle lanes is reduced. The goal of agent is to drive at the desired speed as much as possible and change the lane in time to successfully pass. On one hand, by predicting the future trajectories of surrounding vehicles, their dynamics in advance can be known and judgments can be made in time. On the other hand, by formulating relevant safety rules and make the reward function more comprehensive, the safety of the autonomous vehicle has been greatly improved. In future work, the driving environment is hoped to be more complex and the method proposed in this paper can be advanced optimized .

REFERENCES

[1] Subramanya Nageshrao, H. Eric Tseng, and Dimitar Filev, "Autonomous Highway Driving using Deep Reinforcement Learning," IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, pp. 2326-2331.

[2] Songan Zhang, Huei Peng, Subramanya Nageshrao, and H. Eric Tseng, "Discretionary Lane Change Decision Making using Reinforcement Learning with Model-based Exploration," IEEE International Conference on Machine Learning and Applications (ICMLA), 2019, pp. 844-850.

[3] Konstantinos Makantasis, Maria Kontorinaki1,2, and Ioannis Nikolos, "A Deep Reinforcement-Learning-based Driving Policy for

Autonomous Road Vehicles," Intelligent Transport Systems(IET), 2019.

[4] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J. Kochenderfer, "Safe Reinforcement Learning with Scene Decomposition for Navigating Complex Urban Environments," IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 1469-1476: IEEE.

[5] Zhong Cao, Diange Yang, Shaobing Xu, et al, "Highway Exiting Planner for Automated Vehicles Using Reinforcement Learning," IEEE Transactions on Intelligent Transportation Systems, 2019.

[6] Elfwing S , Seymour B, "Parallel reward and punishment control in humans and robots: safe reinforcement learning using the MaxPain algorithm," IEEE International Conference on Development & Learning & on Epigenetic Robotics. IEEE, 2017.

[7] Branka Mirchevska, Christian Pek Moritz Werling, Matthias Althoff, and Joschka Boedecker, "High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning," International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2156-2162.

[8] Teng Liu, Zejian Deng, Xiaolin Tang, et al, "Predictive Freeway Overtaking Strategy for Automated Vehicles Using Deep Reinforcement Learning," National Natural Science Foundation of China.

[9] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[10] V. Mnih et al., "Human-level control through deep reinforcement learning," vol. 518, no. 7540, p. 529, 2015.

[11] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[12] T. Schaul, J. Quan, I. Antonoglou, and D. J. a. p. a. Silver,"Prioritized experience replay," 2015.

[13] Samyeul Noh and Woo-Yong Han, "Collision Avoidance in On-Road Environment for Autonomous Driving," International Conference on Control, Automation and Systems (ICCAS), 2014, pp. 884-889.