THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

*PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,*
*Volume 24, Number 4/2023, pp. 395–408*
DOI: 10.59277/PRA-SER.A.24.4.11

# A HEURISTIC-ACTION-INVOLVED SAFE LANE-CHANGE OF AUTONOMOUS VEHICLES WITH MULTIPLE CONSTRAINTS ON ROAD CENTERLINE AND SPEED UNDER HIGHWAY ENVIRONMENT

Jun CHEN[1,2], Fazhan TAO[1,4], Zhumu FU[1, 3], Haochen SUN[1], Nan WANG[1]

[1] Henan University of Science and Technology, College of Information Engineering, No.263 Kaiyuan Road, Luoyang, 471023, Henan, China.
[2] Luoyang Normal University, College of Physics and Electronic Information, No.6 Jiqing Road, Luoyang, 471022, Henan, China.
[3] Henan University of Science and Technology, Henan Key Laboratory of Robot and Intelligent Systems, No.263 Kaiyuan Road, Luoyang, 471023, Henan, China.
[4] Longmen Laboratory, Luoyang, 471023, Henan, People's Republic of China
Corresponding author: Zhumu FU, E-mail: fuzhumu@haust.edu.cn

**Abstract**. Lane-change (LC) is one of the most important topics in autonomous vehicles (AVs) on highways. To enhance the implementation of effective LC in AVs, this paper proposes a framework based on deep reinforcement learning, which takes into account heuristic actions and multiple constraints related to the centerline of the road and speed, to improve the overall performance of LC in AVs. Firstly, the influence of unreasonable vehicle actions on the algorithm training process is studied. To improve the rationality of the to-be-trained actions, a novel reasonable action screening mechanism is proposed. Secondly, to keep the vehicle on the centerline of the lane and avoid the collision with other vehicles, a method is designed to calculate the center position of the vehicle. Thirdly, a segmented speed reward mechanism is proposed to constrain vehicle speed. Subsequently, a dynamic reward function is established to train the control algorithm. Lastly, the proposed strategy is evaluated in two simulation scenarios of highways. The simulation results show that the proposed method can increase the number of reasonable actions by more than 30% and improve the success rate of obstacle avoidance with the increase of over 52% in both static and dynamic scenarios compared with the benchmark algorithms.

***Key words:*** lane-change, reasonable actions, constraints, autonomous vehicles, deep reinforcement learning.

## 1. INTRODUCTION

Autonomous vehicles (AVs) can effectively alleviate traffic congestion, reduce the number of accidents, and improve traffic efficiency. However, according to the report published by the National Highway Traffic Safety Administration (NHTSA) [1], there are about 3 million injuries and more than 50 thousand deaths caused by car accidents per year, such as misperception (41%), irrational decision (33%), and improper operation (11%) [2–3]. Therefore, how to improve AV safe driving in high-risk driving scenarios, especially in highway environments, remains a challenge.

The most important factor affecting driving safety is lane-change (LC), and the integration of linear and nonlinear control techniques, along with the exploration of PID control, fuzzy control, and model predictive control (MPC) for unmanned driving research, can indeed provide valuable insights [4, 5]. A lot of papers [6–9] have adopted the method of machine learning to carry out LC decision-making in recent years. The learning-based control method can adapt to the changes of the environment in the training process with constraints and does not depend on the initial model and parameters of the system. The adaptability of the controller based on non-learning to the variable driving scene is limited, and it depends on the parameter selection of the controller. Therefore, learning-based vehicle control methods with constraints have received

increasing research. In [6], an LC predictor based on an adaptive fuzzy neural network was proposed to predict the turning angle by fusion of vehicle sensor information, greatly improving the prediction accuracy of vehicle lane change. [7] proposed a Bayesian network model combined with a Gaussian mixture model to estimate the LC intent in different scenarios. [8, 9] used a deep belief network, support vector machine, and long-short-term memory to model the LC process which consisted of LC decision and LC implementation. However, the above machine learning-based approaches strongly rely on large amounts of training and testing data, which is time-consuming and expensive, especially in collecting the crushed data.

With the development of artificial intelligence, deep reinforcement learning (DRL) brings remarkable advantages to the decision-making of AVs in complex scenarios [10, 11]. The purpose of DRL is to learn and improve the driving experience through continuous trial-and-error, which means that it can be used to help autonomous vehicles avoid collisions [9, 12]. Taking into account the importance of safety in a real driving environment, more and more researchers have introduced safety problems into the training of DRL. In [15], a DRL algorithm with a safe action set technique was employed in decision-making that was effectively coupled to a specially designed trajectory planning model. [16] adopted the safety rule, the safety prediction module, the traumatic memory, and the dynamic potential reward function to further improve the safety and accelerate the learning of LC behavior. In [17], a direction planning based on the conditional depth Q network was proposed to improve the predictive stability of different motion instructions using End-End automatic driving. From the previous study of DRL-based control of unmanned vehicles, safety is the primary consideration. In the DNN training process, it will inevitably result in some unreasonable actions, which will greatly waste training time and reduce training efficiency. Existing studies mainly focus on the reward obtained from the interaction between vehicle actions and the environment, but ignore the rationality of vehicle actions in the actual driving environment [16], and ultimately limit the improvement of DNN training performance. Therefore, the action rationality of DNN must be taken into account, which has rarely been studied. Based on this motivation, this paper proposes various constraint methods for DNN output action to ensure the safe driving.

Therefore, in this paper, a heuristic-action-involved action screening method is proposed, which can teach the DQN to perform reasonable actions according to the actual road conditions, reducing ineffective training time and improving the DQN training efficiency. The main contributions of this paper are summarized:

(1) A heuristic action screening mechanism is introduced to improve the training efficiency of DNN. This mechanism helps eliminate unreasonable actions (URA), reducing ineffective training time, and improving the overall learning efficiency.

(2) To achieve a more realistic highway driving environment, segmented speed constraints are adopted in this paper, which reduces unnecessary computational burden and lead to faster simulations without compromising accuracy.

(3) To keep the vehicle driving in the middle of the lane, a calculation method of vehicle center position is presented with fewer calculation parameters while offering promising application potential.

## 2. PROBLEM STATEMENTS

### 2.1 Lane-change

To describe the problem of the LC behavior, a typical LC problem is drawn in Fig. 1. In this scene, the red car is the ego vehicle (EV), and the blue cars are the surrounding vehicles (SVs). In general, the three-lane is the basic scene of LC research, including the main behaviors of LC [19]. During the process of LC, the main purpose is overtaking. Therefore, the main parameters are the speed and relative position among the EV and SVs [20]. Since the SV behind the EV has little influence on the LC process of the EV, the influence of the SV behind the EV is not considered in this paper.

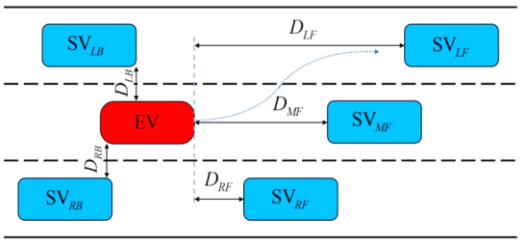


Fig. 1 – The lane-changing scene.

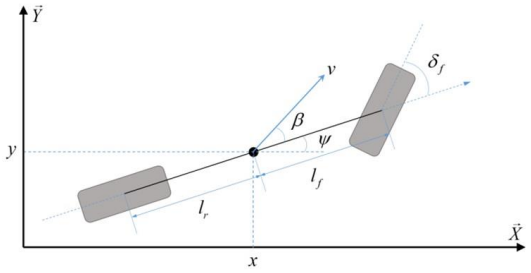


Fig. 2 – Kinematic bicycle model of the vehicle

### 2.2. Vehicle kinematics

To reduce the simulation complexity of this study, the impact of external forces on the vehicle is not considered. Therefore, according to [21], this paper adopts the kinematic model to simulate vehicle movement. The control of steering and speed will act on this model to guide the vehicle's motion behavior. The classical equivalent two-wheel model is shown in Fig. 2. The corresponding kinematic model is [22]:

$$\dot{x} = v\cos(\psi + \beta) \tag{1}$$

$$\dot{y} = v\sin(\psi + \beta) \tag{2}$$

$$\dot{\psi} = \frac{v}{l_r}\sin\beta \tag{3}$$

$$\dot{v} = a_v \tag{4}$$

$$\beta = \arctan\left(\frac{l_r}{l_r + l_f}\tan\delta_f\right) \tag{5}$$

where $x$ and $y$ are the coordinates of the centroid of the vehicle model, respectively, $v$ denotes the velocity of the vehicle, $\psi$ denotes the inertial heading, $l_f$ and $l_r$ denote the distance between the centroid and the center of the front wheel and rear wheel, respectively, $\beta$ denotes the angle between the velocity $v$ of the centroid and the longitudinal axis of the vehicle, $a_v$ is the acceleration of the vehicle, $\delta_f$ is the angle between the wheel and the longitudinal axis of the vehicle, which determines the driving direction of the vehicle.

## 3. PREVIOUS SOLUTIONS

In LC decision-making, the EV selects a series of actions to interact with the environment and then learns whether the action is suitable or not according to the feedback of the environment, to maximize the cumulative reward of an episode. This process is described by the Markov Decision Process (MDP)

$$M = \{S, A, P, R, \gamma\} \tag{6}$$

where $S$ denotes a finite set of states, $A$ denotes a finite set of actions, $P$ denotes the state transition probability, $R$ denotes the reward, and $\gamma$ denotes the discount factor.

The MDP problem is transformed into how to find the optimal policy $\pi^*(s,a)$ in (7) to maximize the expected state-action value of the EV in continuous interaction with the road environment. As long as this value converges, the optimal policy is obtained. The following content uses three kinds of DQN to solve the MDP process:

$$\pi^*(s)=\underset{\pi}{\operatorname{argmax}}\,\mathbb{E}_{\pi}\left[\sum_{i=0}^{\infty}\gamma^i r_{t+i}\mid s_t=s\right] \tag{7}$$

where $\gamma\in[0,1]$ represents the discount of each $r_{t+1}$.

### 3.1. Nature DQN (NDQN)

The NDQN consists of two neural networks, one is the current network $Q_{\text{current}}$ and the other one is the target network $\hat{Q}_{\text{target}}$ [23]. The $Q_{\text{current}}$ network outputs Q values and then uses the $\varepsilon$ greedy algorithm to select actions. The original state, action, reward, and the next state (i.e. $\{s,a,R,s'\}$) will be stored in the experience pool. The $Q_{\text{current}}$ and $\hat{Q}_{\text{target}}$ output the current $Q(s,a,\omega)$ and the target Q value $y_j$, respectively, which are utilized to provide the temporal difference (TD) error. The loss function is defined as

$$J=\mathbb{E}_{\{S,A,R,S'\}\sim M}\left(y_j-Q(s,a,\omega)\right)^2 \tag{8}$$

$$y_j=R+\gamma\max_{A'}\hat{Q}(s',a',\theta) \tag{9}$$

where $\omega$ denotes the weights of $Q_{\text{current}}$, $\theta$ denotes the weights of $\hat{Q}_{\text{target}}$.

### 3.2. Double DQN (DDQN)

In the algorithm of NDQN above, actions are selected by the $\varepsilon$ greedy algorithm and evaluated by the maximum method using (9). Naturally, the overestimation of the Q value will occur to a large extent [24]. In DDQN, the action $\hat{a}$ is selected from the $\arg\max_a Q(s',a',\omega)$, and the target Q value $\hat{Q}_{\text{target}}$ is obtained by

$$y_j=R+\gamma\hat{Q}\left(s',\ \arg\max_{a'}Q(s',a',\omega),\theta\right) \tag{10}$$

which avoids the overestimation of the Q value to some extent, and is also the biggest improvement from DQN to DDQN. The corresponding loss function is

$$J=\mathbb{E}_{\{S,A,R,S'\}\sim M}\left(y_j-Q(s,a,\omega)\right)^2 \tag{11}$$

### 3.3. Dueling DQN (DuDQN)

Traditionally, the DQN uses a single stream output to represent a Q value for each action in a state. The structure by which to directly estimate the action values of an individual action is not very efficient, since the actions are usually relevant and may have similar values. Instead, it is better to first estimate the state values and the relative advantages of every action. The DuDQN decomposes the output Q value into a state value and an action advantage value, which can show whether a certain action is better or worse than other actions. This dual structure is denoted as

$$Q(s,a,\theta)=V(s,\theta)+A(s,a,\theta) \tag{12}$$

where $V(s,\theta)$ denotes the value of the state $s$, and $A(s,a,\theta)$ denotes the advantage value after taking the action $a$ in the state $s$. In DuDQN, [25] utilized a centralized process to implement an important constraint, $\mathbb{E}_{a\sim\pi(s)}\left[A(s,a,\theta)\right]=0$, which is employed to solve unidentifiable problems. After this treatment, the sum of all action advantages in the state is 0, and (12) can be rewritten as

$$Q(s,a,\theta)=V(s,\theta)+\left(A(s,a,\theta)-\frac{1}{|\mathbb{A}|}\sum_{a}A(s,a,\theta)\right) \tag{13}$$

where $|\mathbb{A}|$ denotes the size of the action space.

### 3.4. Parameters calculation

The specific parameter calculation process is as follows: the program first initializes the parameters of the DQN, and the DQN outputs an action for the vehicle to execute. If this action does not cause the vehicle to crash, the DQN will receive a positive reward. If a collision occurs, the DQN will be penalized (negative reward). Then, the DQN neural network is updated by backpropagating the network parameters based on rewards or punishments. The output of the correct actions is reinforced, while the output of the incorrect actions is weakened. The process of updating the network parameters is repeated until the vehicle can avoid all obstacles. This process is automatically completed in the program code.

## 4. PROPOSED METHOD WITH CONSTRAINT ANALYSIS

### 4.1. Reasonable action constraint

The actions output by a DQN in the training phase are random and may sometimes conflict with the actual driving rules of the road. The RA meets the actual driving rules of the road, and the URA breaks the rules. For example, when the vehicle is in the leftmost lane, if the DQN outputs the action of "turning left", it is considered a URA. Similarly, when the vehicle is in the rightmost lane, the action "right turn" is a URA, as shown in Fig. 3. The URA does not cause any change to the environment, but it will result in an ineffective training process of the DQN. Furthermore, in a real highway driving environment, these actions can cause the vehicle to collide with the guardrail, causing many serious accidents. Therefore, these actions are not allowed in practice.

The previous literature [9, 10, 12, 14] has not considered the effect of the URA of the DQN, and the DQN is only trained to reduce collisions by penalizing the final accident result. As a result, these methods have low efficiency in the training of RA, failing to effectively suppress the output of the URA and reduce the training efficiency of the DQN. The aim of the RA constraint designed in this paper is to filter actions heuristically according to actual road constraints during driving before the interaction between the action and the environment, which can avoid the invalid training caused by the URA to improve the training efficiency.

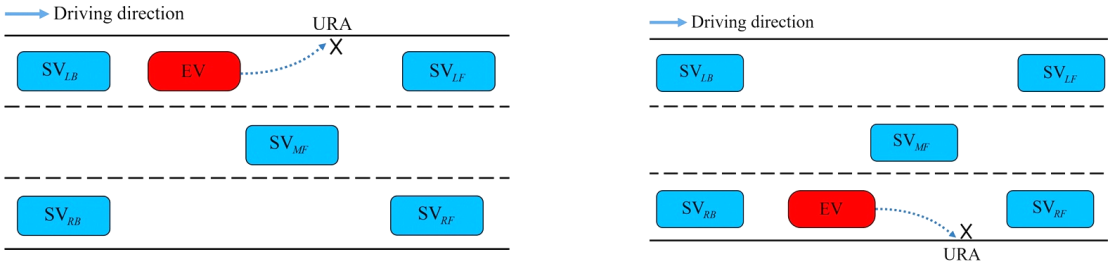


Fig. 3 – Constrain of the URA,

When the DQN outputs a URA, we will negatively reward (penalty) the URA, so that the DQN gradually learns to output as few URA as possible. On the contrary, we positively reward the RA to refine the ‘nice choice’. The reward function for the action is set as

$$R_{\mathrm{ACT}}=\mathrm{Flag}_{\mathrm{ACT}}\times\mathrm{reward}_{\mathrm{ACT}} \tag{14}$$

where $R_{\mathrm{ACT}}$ denotes the reward of the action, $\mathrm{reward}_{\mathrm{ACT}}$ denotes the reward of RA or URA action, generally being a number between 0 and 1, $\mathrm{Flag}_{\mathrm{ACT}}$ is a switching variable being +1 for RA and –1 for URA. The default value of $\mathrm{Flag}_{\mathrm{ACT}}$ is 0.

### 4.2. Speed constraint

In an actual highway driving environment, the speed limit is an inevitable constraint. To prevent vehicles from driving too slowly and causing congestion, a negative reward is given when the speed is less than the minimum speed limit on the highway. To encourage vehicles to drive as fast as possible within the safe speed range and improve traffic efficiency, a positive reward is given when vehicle speed reaches between the lower and upper limits of speed. To ensure safe driving, when the vehicle is overspeeding, a negative reward is generated. The speed reward function is shown as

$$R_{\text{spd}} = \begin{cases} -\text{reward}_{\text{L\_spd}} \times v_{\text{EV}} \\ \text{reward}_{\text{H\_spd}} \times v_{\text{EV}} \\ -\text{reward}_{\text{over\_spd}} \times v_{\text{EV}} \end{cases} \tag{15}$$

where $R_{\text{spd}}$ denotes the reward of speed, $\text{reward}_{\text{L\_spd}}$, $\text{reward}_{\text{H\_spd}}$, and $\text{reward}_{\text{over\_spd}}$ denote the rewards for low speed ( $v_{\text{EV}} < \text{SPD}_{\text{low}}$ ), high speed ( $\text{SPD}_{\text{low}} \le v_{\text{EV}} \le \text{SPD}_{\text{high}}$ ), and overspeed ( $v_{\text{EV}} > \text{SPD}_{\text{high}}$ ), respectively, $v_{\text{EV}}$ denotes the speed of the EV, $\text{SPD}_{\text{low}}$ and $\text{SPD}_{\text{high}}$ denote the lower bound speed and upper bound of speed.

### 4.3. Center position constraint

Generally speaking, human drivers tend to drive close to the center of the road to avoid collisions with vehicles in other lanes. [26] used a soft penalty method to calculate the reward value of the distance between the vehicle and the sidelines of the lane. [27] utilized the center position and the boundary area to calculate the reward for the center position of the vehicle. The above two methods separately calculate the distance between the centroid of the vehicle and the road edge, and the distance between the centroid of the vehicle and the center of the road, which are too complex and not efficient enough.

This paper presents a method for calculating the reward of the EV position directly in Fig. 4, which avoids the calculation of the EV center position and the lane boundary [28]. The calculation process can be expressed as

$$R_{\text{lane}} = -\left( \frac{d_{\text{EV}}^{\text{L}} - d_{\text{EV}}^{\text{R}}}{W_{\text{lane}}} \right)^2 \tag{16}$$

where $R_{\text{lane}}$ is the reward for a position, being negative, $d_{\text{EV}}^{\text{L}}$, and $d_{\text{EV}}^{\text{R}}$ are the distances between the centroid of the EV and the left and right lane boundaries, $W_{\text{lane}}$ is the width of the road. Typically, when the reward $R_{\text{lane}}$ is zero, it means that the vehicle is driving in the middle of the lane.

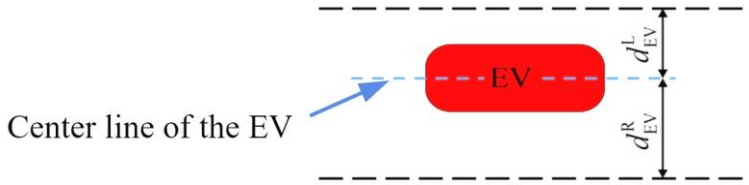


Fig. 4 – Constrain of the middle position of the vehicle.

### 4.4 Safe constraint

A vehicle collision will trigger the termination of a training episode and must be avoided during an actual driving process. In the simulation, the vehicle can rear-end the front vehicle with an acceleration action and may collide with the lateral vehicle under the steering action [29, 30]. The final state of the DQN training is that the EV can safely pass all SVs by changing lanes as fast as possible and without collision. If there is a collition, a negative reward is written as

$$R_{\text{crash}} = -\text{Flag}_{\text{crash}} \times \text{reward}_{\text{crash}} \tag{17}$$

where $R_{\text{crash}}$ denotes the reward for safe driving, $\text{Flag}_{\text{crash}}$ is a switching variable, being 0 for safe driving and -1 for a crush, $\text{reward}_{\text{crash}}$ denotes the reward of a crush.

## 5. DQN-BASED LC DECISION MAKING

The main process of the proposed DQN-based LC decision-making is shown in Fig. 5. The action of the DQN output is judged whether reasonable or not through the constraints, which are shown in the rightmost bold rectangle. The constraints for the highway environment are shown in the leftmost bold rectangle. The RA gets the positive reward, and the vehicle will execute the RA in the environment, while the URA is punished by giving a negative reward, and the vehicle will not execute the URA in the environment.

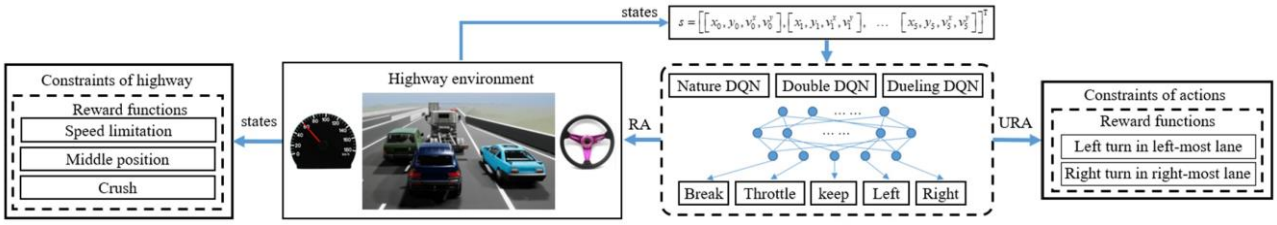


Fig. 5 – The framework of our proposed methods.

### 5.1. State space and action space

The networks of three kinds of DQN algorithms discussed in Section 3 are realized by the fully connected neural network, which has 24 input neurons, 1024 middle layer neurons, and 5 output neurons, and the activate function is Relu. The initialization parameter of the network is a Gaussian distribution with 0 mean value and 0.1 variance. The $\varepsilon$ greedy algorithm is used to select the output action of the network. The state space is the coordinate positions, longitudinal, and lateral velocities of the EV and five SVs, which is represented by

$$\mathbf{s} = \left[\left[x_0, y_0, v_0^x, v_0^y\right], \left[x_1, y_1, v_1^x, v_1^y\right], \quad \cdots \quad \left[x_5, y_5, v_5^x, v_5^y\right]\right]^{\mathrm{T}} \tag{18}$$

where $x_i$, $y_i$ represents the coordinate position of the vehicle, $v_i^x$, $v_i^y$ denote the longitudinal and lateral velocities of the vehicle, $i = 0,1,\cdots 5$. The dimension of matrix s is 6 rows and 4 columns.

The action space represents the result of the DQN output. The EV can use left turn and right turn combined with acceleration and deceleration to act on the LC maneuver. If the opportunity of the LC is not appropriate, the EV will keep the state as it is. The action space is written as

$$A = \{B,\ T,\ K,\ L,\ R\} \tag{19}$$

where *A* denotes the action space, *B*, *T*, *K*, *L* and *R* denote the five actions: break, throttle, keep, turn left, and turn right, respectively.

### 5.2. Reward function

Based on the rewards designed in Section 4, the reward function of this paper can be written as

$$R = R_{\text{ACT}} + R_{\text{lane}} + R_{\text{spd}} + R_{\text{crash}} \tag{20}$$

To ensure that the reward function itself is bounded, the value of *R* is linearly mapped to (0,1) by the equation

$$R' = b_{\min} + \frac{(R - R_{\min})}{(R_{\max} - R_{\min})} b_{\max} \tag{21}$$

where $R'$ represents the modified reward value, ($b_{\min}$ , $b_{\max}$ ) denotes the target range, being (0,1), and ($R_{\min}$ , $R_{\max}$ ) represents the actual range of $R$ in (20).

## 6. EXPERIMENTS ENVIRONMENT

The simulation environment studied in this paper is a physical simulation environment based on Python combined with the highway automatic driving simulation environment (Highway_env) [31], as shown in Fig. 6. The green vehicle is the EV, and the blue vehicles are the SVs. They all emerge at random positions on the three lanes at the beginning of a new episode. The write arrow denotes the driving direction, and the goal of the training is to ensure that the EV can safely overtake all the SVs without colliding as fast as possible.


(a) Normal driving scenario (b) Lane-change (c)Collision

Fig. 6 – The simulation environment: a) normal driving simulation environment; b) an EV changes lane from left to middle; c) the EV collides a SV from behind. After collision, both EV and SV become red.

Both EV and SVs are modeled using the kinematic model in Section 2.2. The longitudinal movement of EV and SVs is described by the IDM (Intelligent Driving Model), and the lateral LC is described by the MOBIL (Minimizing Overall Braking Induced by Lane Change) model. A conventional used three-lane road is used to set up the vehicle driving environment, accompanied by 19 randomly distributed SVs with a speed range 16 ~ 32 m/s (about 60 ~ 115 m/s) and an initial speed of 22 m/s (about 80 km/h). In this paper, both dynamic and static scenarios are used to conduct experiments.

## 7. RESULTS AND DISCUSSION

In this section, the three different DQN algorithms combined with the constraints discussed in Section 4 are tested. The experiments will be examined in four aspects: reward, RA, speed, and success rate of obstacle avoidance in both static and dynamic scenarios.

### 7.1. Reward

In the field of DRL theory, reward is considered the primary metric to assess the strengths and weaknesses of different strategies. Fig. 7 displays the reward outcomes of three DQN algorithms, obtained through both static and dynamic experiments. The red lines represent the rewards generated from the DQN algorithms that adhere to all constraints (NDQN_C, DDQN_C, and DuDQN_C). The blue lines illustrate the rewards generated from the DQN algorithms that only incorporate speed constraints (NDQN_SPD, DDQN_SPD, and DuDQN_SPD). The green lines indicate the rewards generated from the DQN algorithms that solely employ the RA constraint (NDQN_RA, DDQN_RA, and DuDQN_RA).

Taking the static scenario in Figs. 7a–c for example, the reward curves of NDQN_SPD, NDQN_RA, DDQN_SPD, DDQN_RA, DuDQN_SPD, and DuDQN_RA all converge to around 80, while the reward curves of NDQN_C, DDQN_C, and DuDQN_C are stably converged to 140, 180, and 125, respectively. The results in the dynamic scenario Figs. 7d–f are of the same trends. The reward curves of NDQN_SPD, NDQN_RA, DDQN_SPD, DDQN_RA, DuDQN_SPD, and DuDQN_RA all converged to about 80, while the reward curves of NDQN_C, DDQN_C, and DuDQN_C are stably converged at 120, 150, and 116, respectively. This phenomenon of the

reward value fluctuating in an interval is considered to have reached training convergence in the field of DRL, which is different from the stability of the control system. Because the random activities of SVs in dynamic scenarios increase the training difficulty of the NDQN algorithms, the converged reward values of the NDQN_C in dynamic scenarios are slightly lower than those in static scenarios.

In [17], the projected area of the vehicle in the lane and the distance between the vehicle and the lane center are calculated to constrain the position of the vehicle. Compared with our proposed method, the calculation of the vehicle projection area is very complicated, and it is difficult to ensure the accuracy of the calculation. Therefore, the practicability of this method deserves further study. The vehicle center position algorithm proposed in our paper only calculates the distance difference between the center of the vehicle and the lane lines on both sides, which has the characteristics of a simple calculation and increases the training efficiency. The reward value in our study entered the convergence range after about 800 training rounds. However, the reward value in reference [17] still did not show a convergence trend after 400,000 training rounds. Therefore, our proposed algorithm has advantages in producing more rewards and improving DQN training efficiency.

Reference [31] utilizes the same highway simulation environment as ours, which considered the rewards generated by vehicle collisions, rightmost lanes, and speed, respectively. Constraints such as reasonable actions and vehicle position are not taken into account. Therefore, from the training results, our minimum reward value per round is 80, while the maximum reward value per round in [31] is 68.

Therefore, our proposed algorithm has the advantages of producing more rewards and improving DQN training efficiency and has remarkable training stability and convergence.

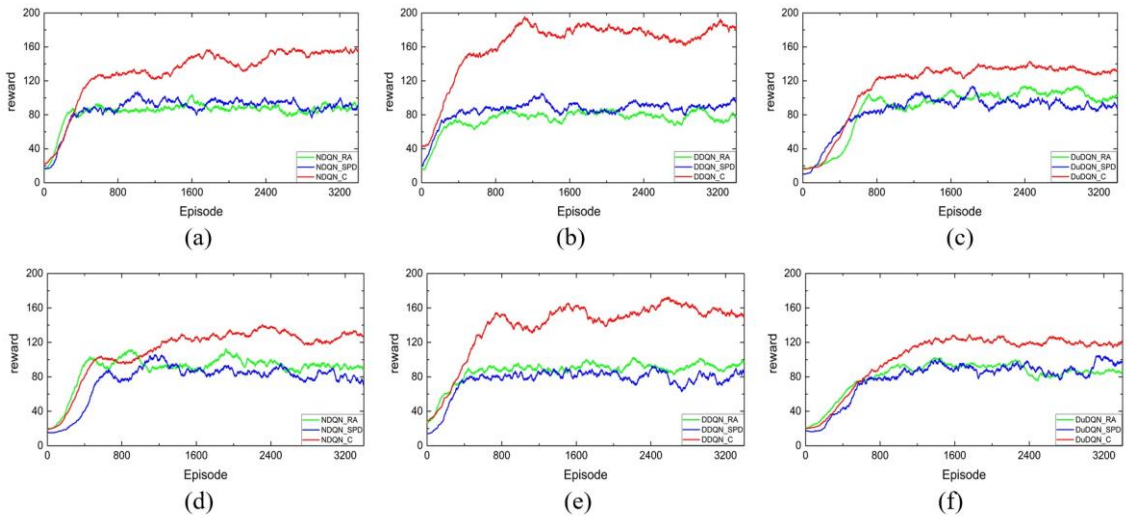


Fig. 7 – Reward comparison diagrams of three DQN algorithms in two scenarios: a)–c) static scenario; d)–f) dynamitic scenario.

### 7.2. Reasonable action

To verify the validity of the action constraints described in Section 4.1, the three DQN algorithms combined with the action constraints are compared in static and dynamic environments, and the results are shown in Fig. 8. In Figs. 8a–f, the horizontal axis represents the training episodes and the vertical axis represents the number of RA of each episode. At the beginning of the training (before 400 episodes), the three original DQN algorithms NDQN, DDQN, and DuDQN (blue lines) and the three proposed DQN algorithms with the RA constraint NDQN_RA, DDQN_RA, and DuDQN_RA (red lines) have the same rising rate.

After about 800 episodes of training, the static scenario in Figs. 8a–c, static scenario, the average numbers of the RA of the NDQN_RA, DDQN_RA, and DuDQN_RA are 80, 82, and 96, respectively, obviously greater than those of the NDQN, DDQN, and DuDQN. In the dynamic scenario in Figs. 8d–f, dynamic scenario, the average numbers of the RA of the NDQN_RA, DDQN_RA, and DuDQN_RA algorithms are 95, 110, and 123, respectively, greater than those of the NDQN, DDQN, and DuDQN. Obviously, in both scenarios, the proposed DQN algorithms with RA constraints can produce more reasonable actions.

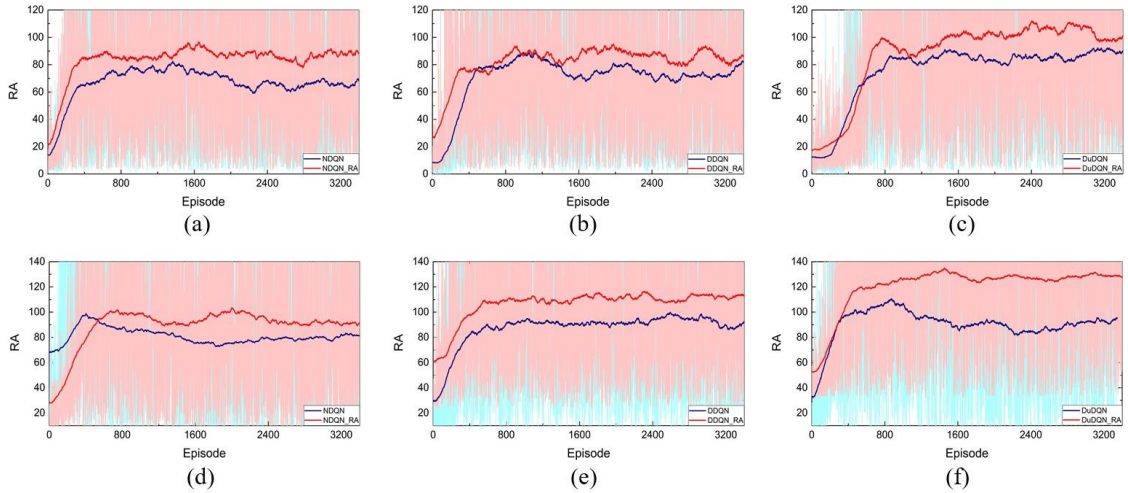


Fig. 8 – RA comparison diagrams of three DQN algorithms in two scenarios with and without RA constraints: a)–c) static scenario; d)–f) dynamitic scenario.

Compared with the three original DQN algorithms in the static scenario, the total number of the RA generated by the NDQN_RA, DDQN_RA, and DuDQN_RA increases from 314956, 308969, and 313927 to 393942, 393826, and 393950, with obvious increases of 25.07%, 27.46%, and 25.49%, respectively. And in the dynamic scenario, the total number of the RA generated by the NDQN_RA, DDQN_RA, and DuDQN_RA increases from 292983, 299045, and 302626 to 393850, 393947, and 394053, with obvious increases of 34.43%, 31.74%, and 30.21%, respectively. The results show that the RA constraint plays an important role in improving the reasonability of the actions, which proves that our heuristic-action-involved method is effective.

### 7.3. Speed

To verify the feasibility of the speed constraints described in Section 4.2, the results of the three DQN algorithms combined with the speed constraints in static and dynamic environments are shown in Fig. 9. Similarly, the horizontal axis represents the training episodes and the vertical axis represents the average vehicle speed of each episode. The red curves indicate the average speeds of the three proposed DQN algorithms with speed constraints, named NDQN_SPD, DDQN_ SPD, and DuDQN_ SPD, and the blue curves indicate the average speeds of the three original DQN algorithms, named NDQN, DDQN, and DuDQN.

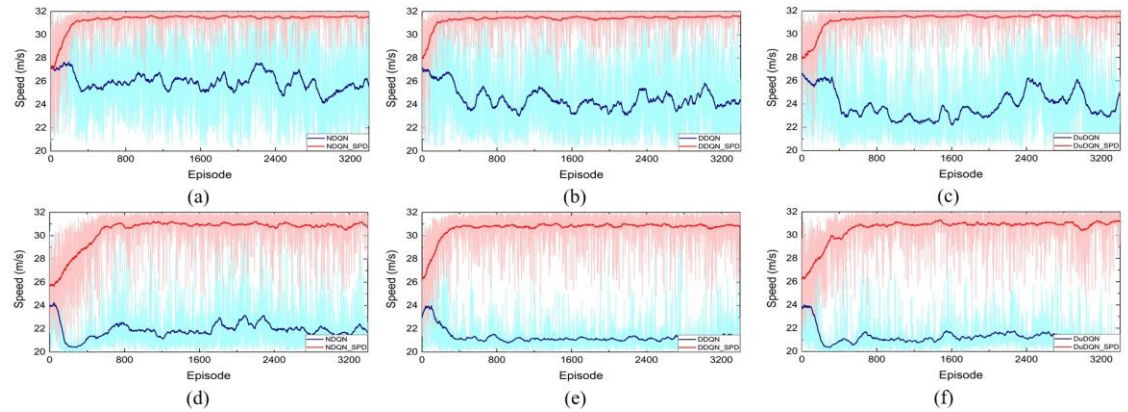


Fig. 9 – Speed comparison diagrams of three DQN algorithms in two scenarios with and without speed constraints: a)–c) static scenario; d)–f) dynamitic scenario.

In the static scene in Figs. 9a–c, the average speeds of the NDQN_SPD, DDQN_SPD, and DuDQN_SPD are 31.31 m/s, 31.35 m/s, and 31.29 m/s, with an improvement of 5.36 m/s, 6.84 m/s, and

7.38 m/s, compared with the three original DQN algorithms, respectively. In the dynamic scene, in Figs. 9d–f, the average speeds of the NDQN_SPD, DDQN_SPD, and DuDQN_SPD are 31.31 m/s, 31.35 m/s, and 31.29 m/s, with the improvement of 5.36 m/s, 6.84 m/s, and 7.38 m/s, compared with three original DQN algorithms, respectively.

The above results can be analyzed as follows. First, the speed reward function (15) designed in this paper can effectively train the EV to drive as fast as possible. Therefore, the final maximum speed of our training was 31.32 m/s, which is greater than 30 m/s in [31]. Second, the stable and fast EV driving indicates that the center position constraint in (16) can avoid collisions to some extent. Because a vehicle collision will immediately terminate the current training episode, which results in a shorter training time and an insufficient acceleration duration, leading to a lower average speed. Third, without the constraints of the center position, the random motion exploration of DQN will produce steering frequently, making the vehicle sway near the center of the lane and greatly reducing the longitudinal speed of the vehicle. All these results indicate that the three original DQN algorithms without speed constraint will make the EV adopt a more and more conservative driving policy to avoid collisions, which is consistent with the results in the existing literature [11, 14].

### 7.4. Success rate in obstacle avoidance

The ultimate purpose of all the above constraints is to improve the vehicle's ability to change lanes and avoid obstacles. The number of successful obstacle avoidance episodes (NSOAE) in the static and dynamic results is listed in Table 1. Since the experimental results in the two scenarios have the same upward trends, only the results in the static state are analyzed here.

The NSOAE of the three proposed DQN algorithms combined with RA constraints only increased by 2.77%, 4.94%, and 9.33% compared with the original DQN algorithms, respectively. This is because the RA constraint only estimates whether an action conflicts with the environment, but does not contribute to the LC and obstacle avoidance.

*Table 1*

The metrics using different DQN in the static scenario

| Method | Static | | Dynamic | |
|---|---|---|---|---|
| | NSOAE | Δ(%) | NSOAE | Δ(%) |
| NDQN | 1014 | -- | 923 | -- |
| DDQN | 931 | -- | 922 | -- |
| DuDQN | 943 | -- | 954 | -- |
| NDQN_RA | 1041 | ↑ 2.77% | 949 | ↑ 2.82% |
| DDQN_RA | 977 | ↑ 4.94% | 952 | ↑ 3.25% |
| DuDQN_RA | 1031 | ↑ 9.33% | 992 | ↑ 3.98% |
| NDQN_SPD | 1520 | ↑ 49.91% | 1435 | ↑ 55.64% |
| DDQN_SPD | 1483 | ↑ 59.29% | 1460 | ↑ 58.35% |
| DuDQN_SPD | 1504 | ↑ 59.49% | 1494 | ↑ 62.34% |
| NDQN_C | 1542 | ↑ 52.07% | 1413 | ↑ 53.09% |
| DDQN_C | 1490 | ↑ 60.04% | 1407 | ↑ 52.61% |
| DuDQN_C | 1454 | ↑ 54.19% | 1470 | ↑ 54.09% |

Because the speed constraint greatly improves the average speed of the vehicle in successful obstacle avoidance episodes, the vehicle can complete each SOAE with fewer training steps. Also, the total number of training steps is fixed; therefore, the DQN algorithm with the speed constraint generates more SOAE. When using NDQN_SPD, DDQN_SPD, and DuDQN_SPD, the NSOAEs increase from 1014, 931, and 943, to 1520, 1483, and 1504, respectively, with an improvement of 49.91%, 59.29%, and 59.49% compared with those of NDQN, DDQN, and DuDQN. Last, the NSOAEs of NDQN_C, DDQN_C, and DuDQN_C increase

to 1542, 1490, and 1454, with an improvement of 52.07%, 60.04%, and 54.19%. compared with those of NDQN, DDQN, and DuDQN.

Furthermore, after adding the center position constraint, it can not only effectively increase the longitudinal speed and reduce the latitude speed, but also control the EV to remain in the middle of the lane, reducing the possibility of collision with SVs in adjacent lanes and improving the NSOAE at the same time.

Because the RA constraint only estimates whether an action conflicts with the environment, the improvements of the NSOAE of DQN algorithms with the RA constraint are not obvious.

Primarily, the integration of all constraints into the DQN algorithms enhances the EV's ability to accurately detect the relative positions and speeds of SVs, thereby enabling the prediction of potential risks in complex environments. Consequently, the model can generate appropriate actions to avoid collisions. The experimental results demonstrate that these novel approaches outperform the original DQN-based methods in both static and dynamic scenarios.

## 8. CONCLUSIONS

In this paper, the constraints-based NDQN, DDQN, and DuDQN algorithms are used to analyze the LC and overtaking behavior of autonomous vehicles. First, the states of the LC and the vehicle dynamics model are clearly described. Second, the action constraint, speed constraint, center position constraint, and collision constraint are proposed. These constraints can help the AV to improve speed and ensure driving safety. Third, the experimental results show that the DQN algorithms combined with all constraints can obtain the largest number of reasonable actions, the fastest speeds, and the highest rewards in both static and dynamic simulation environments.

Although the proposed methods have been validated by experiments, our research still has some limitations as follows.

A more complex distribution of SVs needs to be experimentally verified. In the simulation environment, the SVs kept a certain safe distance. However, in the actual driving environment, this safe distance is not necessarily guaranteed. How to make the vehicle accurately recognize these distance changes is the focus of our future research. The driving style of the driver must be studied. Different driving styles will have different results in the same driving environment. The existing literature [32] has shown that driving style can significantly affect drivers' LC decisions. The driving style is the key research point in the future.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant (62201200), the Program for Science and Technology Innovation Talents in the University of Henan Province under Grant(23HASTIT021), Major Science and Technology Projects of Longmen Laboratory under Grant (231100220200), Aeronautical Science Foundation of China under Grant (20220001042002), the Science and Technology Development Plan of Joint Research Program of Henan under Grant (222103810036). The Scientific and Technological Project of Henan Province under Grant (212102210153, 222102240009).

## REFERENCES

1. S. SINGH, *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*, Traffic Safety Facts-Crash Stats, **2**, pp. 1–2, 2015.
2. M.S. SHIRAZI, B.T. MORRIS, *Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies*, IEEE Transactions on Intelligent Transportation Systems, **18**, *1*, pp. 4–24, 2017.
3. C. YOU, J. LU, D. FILEV, P. TSIOTRAS, *Autonomous planning and control for intelligent vehicles in traffic*, IEEE Transactions on Intelligent Transportation Systems, **21**, *6*, pp. 2339–2349, 2020.
4. M.G. UNGURITU, T.C. NICHITELEA, *Design and assessment of an anti-lock braking system controller*, Romanian Journal of Information Science and Technology, **26**, *1*, pp. 21–32. 2023.

5. H. UCGUN, I. OKTEN, U. YUZGEC, M. KESLER, *Test platform and graphical user interface design for vertical take-off and landing drones*, Romanian Journal of Information Science and Technology, **25**, *3*, pp. 350–367. 2022.
6. J. TANG, F. LIU, W. ZHANG, R. KED, Y. ZOU, *Lane-changes prediction based on adaptive fuzzy neural network*, Expert Systems with Applications, **91**, pp. 452–463, 2018.
7. X.H. LI., W.S. WANG, M. ROETTING, *Estimating driver's lane-change intent considering driving style and contextual traffic*, IEEE Transactions on Intelligent Transportation Systems, **20**, *9*, pp. 3258–3271, 2018.
8. D.F. XIE, Z.Z. FANG, B. JIA, Z. HE, *A data-driven lane-changing model based on deep learning*, Transportation Research Part C: Emerging Technologies, **106**, pp. 41–60, 2019.
9. Y. DOU, F. YAN, D. FENG, *Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers*, Proc. of 19th IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2016, pp. 901–906.
10. C.J. HOEL, K. DRIGGS-CAMPBELL, K. WOLFF, L. LAINE, M-J. KOCHENDERFER, *Combining planning and deep reinforcement learning in tactical decision making for autonomous driving*, IEEE Transactions on Intelligent Vehicles, **5**, *2*, pp. 294–305, 2020.
11. B.R. KIRAN, I.SOBH, V.TALPAERT, P. MANNION, A.A. AL SALLAB, S. YOGAMANI, P. PÉREZ, *Deep reinforcement learning for autonomous driving: A survey*, IEEE Transactions on Intelligent Transportation Systems, **23**, *6*, pp. 4909–4926, 2022.
12. J. WANG, Q. ZHANG, D. ZHAO, Y. CHEN, *Lane change decision-making through deep reinforcement learning with rule-based constraints*, Proc. of 7th International Joint Conference on Neural Networks, 2019, pp. 1–6.
13. B. MIRCHEVSKA, C. PEK, M. WERLING, M. ALTHOFF, J. BOEDECKER, *High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning*, Proc. of 21st International Conference on Intelligent Transportation Systems, 2018, pp. 2156–2162.
14. S. LI, C. WEI, Y. WANG, *Combining decision making and trajectory planning for lane changing using deep reinforcement learning*, IEEE Transactions on Intelligent Transportation Systems, **23**, *9*, pp. 16110–16136, 2022.
15. K.X. LV, X.F. PEI, C. CHEN, J. XU, *A safe and efficient lane change decision-making policy of autonomous driving based on deep reinforcement learning*, Mathematics, **10**, *9*, pp. 1–24, 2022.
16. L. CHEN, X. HU, B. TANG, Y. CHENG, *Conditional DQN-based motion planning with fuzzy logic for autonomous driving*, IEEE Transactions on Intelligent Transportation Systems, **23**, *4*, pp. 2966–2977, 2020.
17. S. NAGESHRAO, E. TSENG, D. FILEV, *Autonomous highway driving using deep reinforcement learning*, Proc. of the Conf. IEEE International Conference on Systems, Man and Cybernetics, 2019, pp. 2326–23331.
18. W. WANG, T. QIE, C. YANG, W. LIU, C. XIANG, K. HUANG, *An intelligent lane-changing behavior prediction and decision-making policy for an autonomous vehicle*, IEEE Transactions on Industrial Electronics, **69**, *3*, pp. 2927–2937, 2021.
19. Y. BIAN, S.E. LI, W. REN, J. WANG, K. LI, H.X. LIU, *Cooperation of multiple connected vehicles at unsignalized intersections: distributed observation, optimization, and control*, IEEE Transactions on Industrial Electronics, **67**, *12*, pp. 10744–10754, 2019.
20. P. POLACK, F. ALTCHE, B. D'ANDREA-NOVEL, A. DE LA FORTELLE, *The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?*, Proc. of 27th IEEE International Conference on Intelligent Vehicles Symposium, 2017, pp. 812–818.
21. J. KONG, M. PFEIFFER, G. SCHILDBACH, F. BORRELLI, *Kinematic and dynamic vehicle models for autonomous driving control design*, Proc. of 25th IEEE International Conference on Intelligent Vehicles Symposium, 2015, pp. 1094–1099.
22. V. MNIH, K. KAVUKCUOGLU, D. SILVER, A.A. RUSU, J. VENESS, M.G. BELLEMARE, A. GRAVES, M.RIEDMILLER, A.K. FIDJELAND, G. OSTROVSKI, *Human-level control through deep reinforcement learning*, Nature, **518**, *7540*, pp. 529–533, 2015.
23. H. HASSELT, A. GUEZ, D. SILVER, *Deep reinforcement learning with double q-learning*, Proc. of 30th International Conference on Artificial Intelligence, 2015, pp. 2094–2100.
24. Z. WANG, T. SCHAUL, M. HESSEL, H. HASSELT, M. LANCTOT, N. FREITAS, *Dueling network architectures for deep reinforcement learning*, Proc. of 33rd International Conference on Machine Learning, 2016, pp. 1995–2003.
25. P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, J. Pan, *Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning*, Proc. 2nd IEEE International Conference on Robotics and Automation, pp. 6252–6259, 2018.
26. G. LI, L. YANG, S. LI, X. LUO, X. QU, G. PAUL, *Human-like decision-making of artificial drivers in intelligent transportation systems: an end-to-end driving behavior prediction approach*, IEEE Intelligent Transportation Systems Magazine, **14**, *6*, pp. 188–205, 2021.
27. J.Q. WANG, J. WU, X.J. ZHENG, D.H. NI, K.Q. LI, *Driving safety field theory modeling and its application in pre-collision warning system*, Transportation Research. Part C: Emerging Technologies, **72**, pp. 306–324.
28. J. ZHANG, G. LU, H. YU, *Effect of the uncertainty level of vehicle-position information on the stability and safety of the car-following process*, IEEE Transactions on Intelligent Transportation Systems, **23**, *6*, pp. 4944–4958, 2020.
29. D. LI, A. LIU, *Personalized lane change decision algorithm using deep reinforcement learning approach*, Applied Intelligence, **53**, *11*, pp. 13192–13205, 2022.
30. A. MAVROGIANNIS, R. CHANDRA, D. MANOCHA, *B-GAP: Behavior-rich simulation and navigation for autonomous driving*, IEEE Robotics and Automation Letters, **7**, *2*, pp. 4718–4725, 2020.
31. F. LI, J. WAGNER, Y. WANG, *Safety-aware adversarial inverse reinforcement learning for highway autonomous driving*. Journal of Autonomous Vehicles and Systems, **4**. pp. 1–13, 2022.

32. G. LI, S.E. LI, B. CHENG, P. GREEN, *Estimation of driving style in naturalistic highway traffic using maneuver transition probabilities*, Transportation Research. Part C: Emerging Technologies. **74**, pp. 113–125, 2017.

*Received May 25, 2023*