2024-01-7035 Published 30 Oct 2024

Interaction-Aware Deep Reinforcement Learning Approach Based on Hybrid Parameterized Action Space for Autonomous Driving

Zhuoren Li, Guizhe Jin, Ran Yu, Bo Leng, and Lu Xiong Tongji University, School of Automotive Studies

Citation: Li, Z., Jin, G., Yu, R., Leng, B. et al., "Interaction-Aware Deep Reinforcement Learning Approach Based on Hybrid Parameterized Action Space for Autonomous Driving," SAE Technical Paper 2024-01-7035, 2024, doi:10.4271/2024-01-7035.

Received: 30 Jul 2024 Revised: 11 Sep 2024 Accepted: 25 Oct 2024

Abstract

Learning-based motion planning methods such as reinforcement learning (RL) have shown great potential of improving the performance of autonomous driving. However, comprehensively ensuring safety and efficiency remain a challenge for motion planning technology. Most current RL methods output discrete behavioral action or continuous control action, which lack an intuitive representation of the future motion and then face the problems with unstable or reckless driving behavior. To address these issues, this work proposes an interaction-aware reinforcement learning approach based on hybrid parameterized action space for autonomous driving in lane change scenario. The proposed method can output high-level feasible trajectory and low-level actuator control command to control the vehicle's motion together. Meanwhile, the reward functions for the local traffic environment are designed to evaluate the effect of the interaction between ego vehicle and surrounding vehicles. The contributions of the proposed method are: 1) propose a hybrid parameterized action based interaction-aware DRL framework (*HPA-IDRL*); 2) the proposed *HPA-IDRL* can learn from the reward not only considering self-benefits but also considering the benefits of the local traffic environment; 3) A multi-head attention layer is embedded before actor network and critic network respectively to exploit the interactive information in the traffic environment. Thus, the *HPA-IDRL* agent can generate more flexible and smooth driving behavior, which improves the safety and the efficiency of autonomous driving. The proposed method is implemented and validated with other four advanced DRL model in various simulation environments. The results demonstrate that the proposed *HPA-IDRL* can effectively balance the flexibility and smoothness of driving behavior, leading to the improving performance that is both safe and efficient.

Keywords

Autonomous driving, motion planning, reinforcement learning

Introduction

Autonomous driving technology has good potential to improve driving safety and traffic efficiency [1, 2]. Some Robotaxi and Robobus products have been deployed on public roads, such as Baidu-Apollo in Wuhan, Changsha, and Waymo in San Francisco, and others. However, according to corresponding reports, many takeover incidents are still recorded and the autonomous vehicles being criticized for causing traffic jams by driving slowly and stopping unexpectedly. [3, 4] Therefore, autonomous driving technology still has a long way to go in terms of safety and flexibility.

Motion planning, broadly defined to include behavioral decision-making, trajectory planning, and motion control, is regarded as the brain of autonomous driving [5]. The results of motion planning directly determine the intelligence of the autonomous driving system [6]. The motion planning module receives perception information and enables autonomous vehicle to make corresponding motion maneuvers, which makes a significant impact on the safety, efficiency and comfort performances of autonomous vehicles [7].

Traditional rule-based techniques had played an important role in motion planning of autonomous driving technology [2]. They describe behavior models intuitively according to vehicle motion models, traffic rules and driving experience [8].

Early rule-based motion planning methods separate behavioral decision-making and trajectory planning completely. The Decision-making approaches include Finite State Machines (FSM) [9], Behavior Tree (BT) [10] and Markov Decision Process (MDP) [11], etc. They often generate the semantic behavior and then plan the feasible trajectory through curve-based or sample-based methods [12, 13]. Their models are simple to construct but have limited application conditions. Optimization methods, such as Model Predictive Control (MPC), have subsequently been widely used to integrate decision making with trajectory planning and motion control [14]. Artificial Potential Fields (APF) can be well adapted to MPC framework to provide a reference for design of objective function and constraint [14]. In [5], the authors use Game Theory to generate high-level semantic behaviors, which are provided to MPC problem to solve the low-level control command. [15] combines discrete lane with continuous vehicle kinematics model, and then construct a hybrid MPC problem to simultaneously solve semantic behavior and motion control command.

In recent years, learning-based motion planning method represented by imitation learning (IL) and reinforcement learning (RL) are widely studied. These methods can learn a complex driving policy from driving data, and have been become a highly promising paradigm for autonomous driving. IL can directly fit the optimal driving strategy distribution from the dataset [16], while it needs large amount of expert data and still face distribution shift problems [17]. Unlike IL, RL agent generates policies by interacting with the environment and evaluating itself with the reward function, which not rely on marked data and allows the performance to exceed human-level [18]. This method is modeled based on MDP with long-term rewards to construct the task of autonomous driving in complex environments. Deep reinforcement learning (DRL) combined with deep neural networks (DNN) has excellent nonlinear approximation capability and can generate intelligent policy in a model-free manner [19]. DRL motion planning methods has been tried and achieved great effects in many scenarios such as lane changing, merge, intersection, etc. [20, 21]. However, most current DRL motion planning methods frequently encountered problems including unstable action output and unsafe maneuvers, which may lead to uncomfortable driving experience and even collision accident.

On one hand, the previous work involved two main types of action space design approaches, including discrete semantic behavior actions and continuous actuator control command actions [22]. They both directly utilize the RL's output action to control the vehicle. For discrete semantic behavior, these actions have a limited effect on vehicle's maneuvering because there are planning and control module after behavior decision. For continuous actuator control command, it is easy to lead the fluctuation of the vehicle's motion since it directly controls the vehicle. Both two classes methods lack an intuitive representation of future motion, which has unstable action output and prevents full confidence in current DRL output actions [23]. Some work attaches rule-based planning/control methods after the DRL output action, such as DQN method, to generate feasible path and then track it [24]. However, this method actually generates the candidate trajectory set based on a finite discrete action space, which loses the flexibility of the DRL policy and thus may lead to overly conservative maneuvers, which defeats the original purpose of using DRL to solve autonomous driving problems [25]. The above action architecture lack the detailed analysis and implementation combination of the vehicle's driving behavior. Therefore, it is always challenging for them to comprehensively improve the driving safety and efficiency.

On other hand, DRL learn the optimal policy according to the long-term reward to generate smarter behaviors. Since RL agent focuses on maximizing the reward function, it is likely to explore unsafe behaviors during the learning process and even after training [26]. It is hard for RL agent to simultaneously learn safety and other goals such as efficiency through a single reward function [27]. In addition, focusing only on own safety rewards and ignoring the impacts that the ego vehicle brings to the surrounding environment also tends to increase the overall potential risk and reduce the travel efficiency of the local traffic environment [28]. Although DRL implicitly models the interaction mechanism between the ego vehicle and the surrounding vehicles, it is still difficult to learn it well based only on directly observed states and the egoist reward function. This indirect unsafety is not currently well considered in the frameworks of single agent DRL, which makes it difficult for RL agent to understand the environment changes well, and makes them prone to overly aggressive or conservative behavior.

To address these problems above, this paper proposes an interaction-aware reinforcement learning approach based on hybrid parameterized action space for autonomous driving. The focus of this work is to make DRL agent generate flexible and safe maneuvers in lane change scenario. It can output high-level feasible trajectory and low-level actuator control command to control the vehicle's motion together. Meanwhile, the reward functions for the local traffic environment are designed to evaluate the effect of the interaction between ego vehicle and surrounding vehicles. The contributions of the proposed method are summarized as following:

1). **DRL with Hybrid Parameterized Action:** We propose a hybrid parameterized action based interaction-aware DRL framework (*HPA-IDRL*). It can output the parameterized action to generate feasible lane change path. Lateral control command through rule-based tracking module and the longitudinal control command output by DRL agent are executed on vehicle system together, which improves the stability of the lateral motion behavior while keeps the flexibility of the output by the DRL agent.

2). **Multi-Critic Learning with Interaction Rewards:** The proposed *HPA-IDRL* can learn from the reward not only considering self-benefits but also considering the benefits of the local traffic environment. Interaction rewards are designed to describe the driving cost of the interactive behavior between ego vehicle and surrounding vehicles. Multiple critic networks are designed to evaluate Q values based on different reward functions, allowing RL agents to better focus on different driving object.

3). **Interaction-Aware Actor-Critic Mechanism:** To make DRL agent better understand the interaction character from the traffic environment, multi-attention mechanism is used to extract the state input feature. A multi-head attention layer is embedded before actor network and critic network respectively to exploit the interactive information in the traffic environment. Thus, the DRL agent can generate more flexible and safer actions.

The remainder of this work is organized as: Section II introduces some algorithm background. In Section III, the proposed *HPA-IDRL* approach is described in detail. Section IV is the specific implementation. And then in Section V, simulation testing and discussion are presented. Finally, Section VI concludes this work.

Background and Problem Definition

Reinforcement Learning

The motion planning problem of autonomous diving in RL is often modeled as a Markov decision process (MDP) by the tuple $\langle \mathcal{S}, \mathcal{A}, \boldsymbol{R}, \mathcal{T}, \gamma \rangle$, where the S is the state space: $s \in \mathcal{S}, \mathcal{A}$ is the action space: $a \in \mathcal{A}, \boldsymbol{R}$ is the reward function based on current state and action: $r \in \boldsymbol{R}, \mathcal{T} = \mathcal{T}(s,a)$ is the system dynamics transiting to next state s' when the RL agent take an action a, and the $\gamma \in (0,1]$ is a discount factor to decay future rewards. The policy $\boldsymbol{\pi}$ is a distribution of over action according to state, i.e. $\boldsymbol{\pi}(s|a)$. The goal of RL is to learn an optimal policy $\boldsymbol{\pi}^*$, which maximizes the long-term expected discounted return $\boldsymbol{R}_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ at every time step t:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\boldsymbol{R}_t\right] = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\Sigma \gamma^t r\left(s_t, a_t\right)\right] \quad (1)$$

It is usually to estimate the state-action-value (Q-value) function by the Bellman Equation to find the optimal policy, which can be defined as:

$$\mathcal{Q}\left(s_t, a_t\right) = r_t\left(s, a\right) + \gamma \mathbb{E}\left[\mathcal{Q}\left(s_{t+1}, a_{t+1}\right)\right] \quad (2)$$

and then the optimal policy $\boldsymbol{\pi}^*$can be obtained by maximizing the $\mathcal{Q}_t(s,a)$. The DRL aims to estimate the max Q-value function $\mathcal{Q}_t^*\left(s,a\right) = \max_{\pi} \mathcal{Q}^{\pi}\left(s,a\right)$ by a DNN with weight parameters θ as $\mathcal{Q}_t\ (s, a; \theta)$[19]. The temporal difference (TD) error

$$\delta_t = r\left(s_t, a_t\right) + \boldsymbol{\gamma} \max_a \mathcal{Q}\left(s_{t+1},; a_{t+1},; \theta'\right) - \mathcal{Q}\left(s_t,; a_t,; \theta\right) \quad (3)$$

is used to optimize the weight parameters of evaluate network (θ) and target network (θ’) by the gradient descent with the loss function $\mathcal{L}_t\left(\theta\right) = \delta_t^2$.

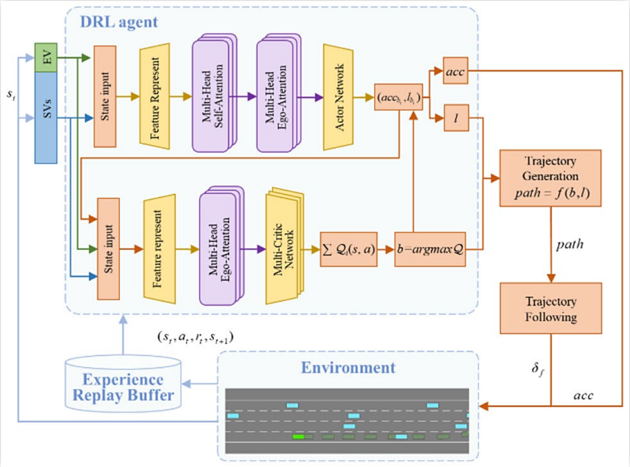
Problem Definition

Considering the DRL motion planning, it would be desirable if the RL agent could provide the trajectory or path of its future motion, although this is really challenging. A natural idea is to let the DRL output action to participate generating trajectory. As mentioned in Introduction Section, candidate trajectory set based on finite discrete RL semantic actions loses the flexibility of the DRL policy.

Focusing on the lane-change maneuver, the lateral lane-changing behavior is usually discrete under structured roads, while the specific length of the lane-changing trajectory varies continuously according to the actual scenarios. Therefore, this work proposes an interaction-aware DRL framework based on hybrid parameterized action (*HPA-IDRL*), in which we design a hybrid action space including: 1) discreate lane-change semantic behavior *b*, 2) continuous lane-change trajectory longitudinal length *l*, and 3) continuous real-time acceleration value *acc*. On this basis, an interaction-aware reinforcement learning approach is design with multi-critic evaluation and multi-head attention mechanism. The DRL agent can better understand the state feature relevance of surrounding vehicles，and better update the policy network through multi-critic using interaction-aware rewards.

The framework of the proposed approach is shown in Figure 1, which mainly consists of the DRL agent and the traffic environment. The DRL agent generates the action from the observed states from ego vehicle and surrounding vehicles. These states are encoded into feature vectors, go through multi-head attention layers and then decoded as the continuous actions(acc_{b_i}, l_{b_i})for all possible discrete action b_i. They are further encoded together with the observed states through attention layers and multi-Critic layers, thereby synchronously generating optimal discrete semantic behavior *b*. Thee parameterized actions (b, l)are used to generate feasible trajectory and further output steering angle δ_f through a rule-based trajectory following module. δ_f and acc are executed on the vehicle system to control the vehicle’s motion, which enhances the stability of lane-change maneuvers to a certain extent while also keep the flexibility of the DRL policy.

FIGURE 1 Conceptual framework of the proposed HPA-IDRL.



Approach Detail

DRL with Hybrid Parameterized Action

The action space is designed as *acc*:

$$\mathcal{A} = \left\{ \left(b,l,acc \right) | l \in L_b, acc \in A_b \text{ for } b \in \left[B \right] \right\} \tag{4}$$

where L_b, A_b, and $[B]$ is the subspace of the action respectively. The Bellman Equation (2) can be written as:

$$\begin{aligned} &\mathcal{Q}\left(s_t, b_t, l_{b_t}, acc_{b_t} \right) = \\ &\mathbb{E}\left[r_t\left(s, b_t, l_{b_t}, acc_{b_t} \right) + \gamma \max_{b \in [B]} \left\{ \sup_{L_b, A_b} \mathcal{Q}\left(s_{t+1}, b, l_{b_{t+1}}, acc_{b_{t+1}} \right) \right\} \right] \end{aligned} \tag{5}$$

The DRL with parameterized action space learns concurrently from four DNNs, including two Actor-networks (evaluate network: $\mu(\theta^\mu)$ and target network: $\mu'(\theta^{\mu'})$) and two critic-networks (evaluate network: $\mathcal{Q}_t\left(\theta^{\mathcal{Q}}\right)$ and target network: $\mathcal{Q}'\left(\theta^{\mathcal{Q}'}\right)$). During the training, it is hope to find a set of θ^μ to maximize the $\mathcal{Q}_t\left(\theta^{\mathcal{Q}}\right)$ for each $b \in [B]$ (such as Eq. (5)). Therefore, the loss functions for networks update process are defined as follows:

$$\begin{aligned} y_t &= r_t + \gamma \max_{b \in [B]} \mathcal{Q}'(s_{t+1}, b, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{\mathcal{Q}'}) \\ \mathcal{L}(\theta^{\mathcal{Q}}) &= \frac{1}{2}\left[y_t - \mathcal{Q}(s_{t+1}, b_{t+1}, \mu(s_{t+1} \mid \theta^\mu) \mid \theta^{\mathcal{Q}}) \right]^2 \\ \mathcal{L}(\theta^\mu) &= -\sum\nolimits_b \mathcal{Q}(s_t, b, \mu(s_t \mid \theta^\mu) \mid \theta^{\mathcal{Q}}) \end{aligned} \tag{6}$$

Thus, the output action (b_t, l_t, acc_t) can be obtained simultaneously by:

$$(b_t, l_t, acc_t) == \underset{b_t \in [B]}{\operatorname{argmax}} \mathcal{Q}(s_t, b_t, \mu(s_t \mid \theta^\mu)_t \mid \theta^{\mathcal{Q}}) \tag{7}$$

Multi-Critic Learning with Interaction Rewards

In the real environment, the action selection of actor-network needs to evaluate multiple object goals. However, there is a coupling of evaluation metrics in a single reward function in one critic-network in typical Actor-Critic DRL, which creates turbulent phenomenon in the training process and leads to inefficient learning [29]. In addition, using a single value function shared over multiple objects can result in negative interference between different objects, which can compromise learning performance [27].

This work proposed multi-Critic Network with interaction reward functions, aiming to effectively guide the iterative updating of the actor-network by evaluating the generated policy from multiple perspectives, including from EV itself and from the surrounding traffic environment. The reward functions are designed as:

$$\begin{aligned} r_t &= \left\{ r_{m,t} \right\}, m \in \left[1, M \right] \\ r_{m,t} &= g\left(EV \right) \text{ or } g\left(EV, SVs \right) \end{aligned} \tag{8}$$

where $r_{m,t}$ is the reward function designed for the m^{th} critic-network, M is the number of critic-networks. In this work, the $r_{m,t}$ includes not only the egoist gain $g(EV)$, but also the interaction-aware gain of the local traffic with surrounding vehicles $g(EV, SVs)$. Therefore, for multiple Q-value functions, the Bellman Equation (5) can be rewritten as:

$$\begin{aligned} &\mathcal{Q}_m(s_t, b_t, l_{b_t}, acc_{b_t} \mid \theta_m^{\mathcal{Q}}) \\ &= \mathbb{E}\left[r_{m,t} + \gamma \max_{b \in [B]} \left\{ \sup_{L_b, A_b} \mathcal{Q}_m(s_{t+1}, b, l_{b_{t+1}}, acc_{b_{t+1}} \mid \theta_m^{\mathcal{Q}}) \right\} \right] \\ &= \mathbb{E}\left[r_{m,t} + \gamma \max_{b \in [B]} \left\{ \mathcal{Q}_m(s_{t+1}, b_{t+1}, \mu(s_{t+1} \mid \theta^\mu) \mid \theta_m^{\mathcal{Q}}) \right\} \right] \end{aligned} \tag{9}$$

$$\mathcal{Q}_{all} = \sum_{m=1}^{M} \omega_m \mathcal{Q}_m(s_t, b_t, l_{b_t}, acc_{b_t} \mid \theta_m^{\mathcal{Q}}) \tag{10}$$

where $\theta_m^{\mathcal{Q}}$ is the network parameter of the m^{th} critic-network, $\mathcal{Q}_m(s_t, b_t, l_{b_t}, acc_{b_t} \mid \theta_m^{\mathcal{Q}})$ is the Q-value of m^{th} evaluate, and ω_m is the corresponding weight of the Q-value. Then, the target value function can be defined as:

$$y_{m,t} = r_{m,t} + \gamma \max_{b \in [B]} \mathcal{Q}'_m(s_{t+1}, b, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta_m^{\mathcal{Q}'}) \tag{11}$$

The corresponding loss functions for multiple critic-networks are as following:

$$\mathcal{L}_m(\theta^{\mathcal{Q}}) = \frac{1}{2}\left[y_{m,t} - \mathcal{Q}_m(s_{t+1}, b_{t+1}, \mu(s_{t+1} \mid \theta^\mu) \mid \theta_m^{\mathcal{Q}'}) \right]^2 \tag{12}$$

Interaction-Aware Actor-Critic Mechanism

In this section, multi-head self-attention and ego-attention [30] layers are embedded in the DRL structure to better exploit the interactive information in the traffic environment.

The Multi-Layer attention mechanism for the actor-network is shown in Figure 2(a). The input vectors are the state features of EV and SVs from a linear encoding layer whose weights are shared between all vehicles. They are then fed to a multi-head self-attention layer, which is composed of several heads stacked together. The output is then fed to a multi-head ego-attention layer, while only the first element $\Phi_{f,1}^{(s)}$ of the output $\Phi_{f,i}^{(s)}$, i.e., the self-attention feature of the EV, emits a single query $\Phi_{Q}^{(e)}$ with a linear projection. The remain elements of $\Phi_{f,i}^{(s)}$ are emitted as the keys and values. The final output $\Phi_{F}^{(e)}$ from all heads is decoded to obtain the mean and variance of Gaussian-distribution for the DRL actions.

As shown in Figure 2(b), for the critic-network, there is only a multi-head ego-attention layer, while the final output is decoded with multiple linear layer to get the Q-value $\mathcal{Q}_m$ of each critic-network.

Specific Implementation

Observation States and Action Design

State In this work, the DRL agent observe the information of the SVs in neighboring lanes and of EV itself by,

$$\mathcal{S} \overset{\Delta}{=} \left[\left\{ p_i, x_i, y_i, \varphi_i, v_{ix}, v_{iy} \right\}_{i=0,\cdots,8} \right] \tag{13}$$

which consists of the flag p_i indicating whether the ith car is observed or not (for EV, p_0 is always equal to 1), the position x_i, y_i and the heading angle φ_i in road coordinate, and the speed v_{ix} ,v_{iy} in lateral and longitudinal directions. i=0 represents the EV and i=1~8 represents the SVs. It is assumed that SVs outside the neighboring lanes are not considered. The DRL agent controlling the EV can observe SVs within the observation range L_{front} = 160 m and L_{back} = 80 m, as shown in Figure 3.

Action At every time step t, the DRL agent choose the discrete action and generate the Gaussian-distribution parameters of continuous action simultaneously. The discrete action space means the lateral semantic lane-change behaviors [B]: {-w_r}: '*left lane change*', {w_r}: '*right lane change*', {0}: '*keep current lane*', where w_r is the road width. The continuous action space includes the longitudinal length $L_b \in [L_{bmin}, L_{bmax}]$ of the desired trajectory and the acceleration $A_b \in [A_{bmin}, A_{bmax}]$.

$$\mathcal{A} = \left\{ \left[B \right], L_b, A_b \right\} \tag{14}$$

With the parameterized action (b, l), the desired trajectory at current time step can be generated as:

$$y_{0,t+k} = \alpha_5 x_{0,t+k}^5 + \alpha_4 x_{0,t+k}^4 + \alpha_3 x_{0,t+k}^3 + \alpha_2 x_{0,t+k}^2 + \alpha_1 x_{0,t+k} \tag{15}$$

FIGURE 3 Environment states and the observation range.

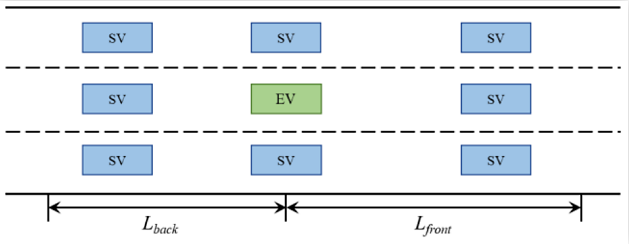
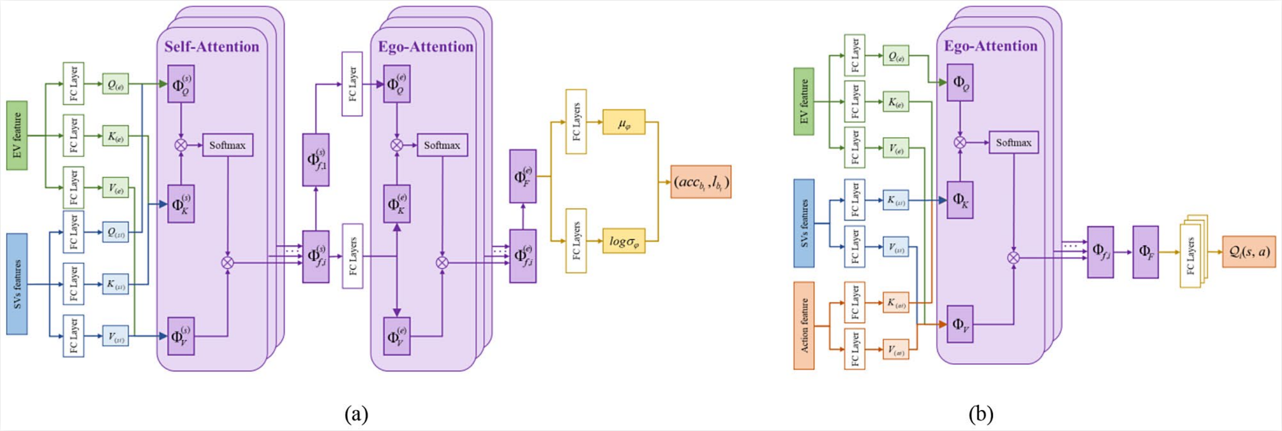


FIGURE 2 Multi-Layer attention mechanism for the actor-Network: (a) multi-attention layer embedded in actor-network, (b) ego-attention for critic-network.



where ($x_{0,\,t+k}$,$y_{0,\,t+k}$) is the position of the waypoint at the *t+k* time step, *k*∈[1, K_p], *Kp* is the planning horizon. The heading angle $\varphi_{0,\,t+k}$ can be obtained through differential algorithm, and the corresponding weight $\alpha_{1\sim5}$ can be calculated according to the position constraints of the current position and the target position:

$$\begin{cases} y_{0,t} = \alpha_5 x_{0,t}^5 + \alpha_4 x_{0,t}^4 + \alpha_3 x_{0,t}^3 + \alpha_2 x_{0,t}^2 + \alpha_1 x_{0,t} \\ y_{0,t+K_p} = \alpha_5 x_{0,t+K_p}^5 + \alpha_4 x_{0,t+K_p}^4 + \alpha_3 x_{0,t+K_p}^3 + \alpha_2 x_{0,t+K_p}^2 + \alpha_1 x_{0,t+K_p} \end{cases} \tag{16}$$

$$\begin{cases} x_{0,t+K_p} = l_t \\ y_{0,t+K_p} = \begin{cases} w, & \text{if } b_t = \text{'}\mathit{left\ lane\ change}\text{'} \\ 0, & \text{if } b_t = \text{'}\mathit{keep\ current\ lane}\text{'} \\ -w, & \text{if } b_t = \text{'}\mathit{right\ lane\ change}\text{'} \end{cases} \end{cases} \tag{17}$$

The acceleration *acc* is executed on the vehicle system with the steering angle δ_f, which is obtained through a rule-based trajectory following module using Stanley model, with the input of the desired trajectory.

Reward Functions

In this work, we construct two critic-networks with two different rewards: egoist reward and interaction-aware reward.

Egoist Reward The goal of the DRL agent is to drive at the desired speed as much as possible without collision. Thus, the egoist reward consists of self-safety reward, the efficiency reward, the comfort reward and the behavioral continuity reward. The self-safety reward focus on whether EV has collision with SVs, i.e. r_{safe} = -1 when collision happened, otherwise r_{safe} =0. The efficiency reward r_{eff} is the is the absolute value of the difference between the v_{0x} and the desired speed v_{des}, i.e. | v_{0x} - v_{des} |. The comfort reward r_{comf} prefers the steering angle δ_f and acceleration *acc* to be as close to zero as possible. At last, there is a behavioral continuity reward r_{con} to punish jittery semantic behavior. As summary, the egoist reward can be defined as follows:

$$\begin{aligned} & r_{1,t} = g_{1,t}\left(EV\right) = 0.5 r_{safe} + 0.3 r_{eff} + 0.1 r_{comf} + 0.1 r_{con} \\ & r_{safe} = \begin{cases} 0, \text{coliision} \\ 1\ \text{,else} \end{cases} \\ & r_{eff} = 1 - \frac{\left| v_{0x} - v_{des} \right|}{v_{des}} \\ & r_{comf} = 1 - \left(0.5 \frac{\left|\delta_f\right|}{\left|\delta_{\max}\right|} + 0.5 \frac{\left|acc_t\right|}{\left|acc_{t\max}\right|} \right) \\ & r_{con} = 1 - \left(b_t - b_{t-1}\right) \end{aligned} \tag{18}$$

Interaction-Aware Reward In a lane-changing maneuver, the EV inevitably affects the rear SV in the target lane, which indirectly affects the behavior of the SVs traveling within the neighborhood in the target lane. In order to consider the impact of the behavior of the EV on the local traffic system (consists of the EV and the SVs in the target lane), EV and SVs are modeled as lane-change game participants inspired by the game theory [5]. The interaction-aware reward function is designed from the perspective of equilibrium of both parties' gains, which also includes the cost of safety and efficiency. The safety cost C_{safe} consists of lateral and longitudinal safety for both EV and SVs:

$$\begin{aligned} & C_{safe} = C_{safe,lat} + C_{safe,lon} \\ & C_{safe,lat} = \sum_{k=0}^{K_{lat}} \left[\kappa_{lat1}\left(v_{0y} - v_{ky}\right)^2 + \frac{\kappa_{lat2}}{\left(y_{0y} - y_{ky}\right)^2} \right] \\ & C_{safe,lon} = \sum_{k=0}^{K_{lon}} \left[\kappa_{lon1}\left(v_{0x} - v_{kx}\right)^2 + \frac{\kappa_{lat2}}{\left(x_{0x} - x_{kx}\right)^2} \right] \end{aligned} \tag{19}$$

where K_{lat} and K_{lon} represent the SVs in two directions. The efficiency cost C_{eff} consists of the efficiency reward of both EV and SVs, such as

$$C_{eff} = \sum_{k=0}^{8} p_i \left| v_{ix} - v_{des} \right| \tag{20}$$

Hence, the interaction-aware reward can be written as:

$$r_{2,t} = g_{2,t}\left(EV, SVs\right) = C_{safe} + C_{eff} \tag{21}$$

Training and Testing Process

The proposed *HPA-IDRL* is trained in a Highway-Env open source simulator [31]. The training environment is a three-lane highway scenario, the SVs are generated randomly on each lane with random initial position and speed with the designed traffic density d_t. Additionally, some advanced DRL baseline model including *DQN* [23], *PPO* [32], *SAC*$_c$ [33] (with continuous command actions of steering angle and acceleration), and *SAC*$_h$ [34] (with hybrid actions of trajectory parameters and acceleration, but it is cut off the continuous action directly to get discrete action). The difference between these five DRL agent is listed in Table 1. All the DRL agents are trained for 10,000 episodes in

TABLE 1 Difference between each DRL agent

	DQN	PPO	*SAC*$_c$	*SAC*$_h$	*HPA-IDRL*
Discrete semantic behavior	✓	×	×	×	✓
Continuous control command	×	✓	✓	✓	✓
Trajectory parameterized action	×	×	×	✓	✓

TABLE 2 Parameters of DRL agent and simulation environment for training process

Parameter	value
Traffic density d_t in training	0.3
Traffic density d_t in testing	0.3 and 0.5
Number of training episodes	1250
Number of Testing episodes	100
Max length of episode	100
Learning rate	0.0001
Discount factor γ	0.8
Road width w_r	3.5 m
Vehicle length	5.0 m
Vehicle width	1.8 m
Policy frequency	10 Hz
Ego-attention head	8
Self-attention head	8
Experience replay buffer size	30,000
Mini-batch buffer size	256
Activation function	ReLU

same environment for 5 times with different random seeds, and then the trained DRL agents are tested in two scenarios: scenario 1 is same as the training (traffic density d_t = 0.3), and scenario 2 is a more congested traffic environment then training (traffic density d_t = 0.5), which can be used to analyze the performance of different agent far from the training scenario. For all DRL agents, the encoder and decode layers of each network are both [64×128×64] units. Some parameters of training and testing process are listed in Table 2.

Results and Discussion

Evaluation Metrics

In this work, the driving goal of EV is to drive in the highway as far as possible and avoid to colliding with SVs. Serval metrics are used to evaluate the performance of the proposed method.

1). **gained reward:** For every DRL agent, the gained reward is always the most frequently used evaluation metric, which comprehensively assesses the agent's performance on a given task

2). **collision rate:** Safety is a fundamental requirement for autonomous driving. The collision rate provides a intuitive measure of how safe an agent is to drive.

3). **average speed:** Driving efficiency is also a noteworthy metric. The average speed, along with the collision rate, can be used to evaluate the agent's intelligence together.

4). **number of lane-change:** This metric gives some indication of the driving flexibility. It can be analyzed together with the average speed to find out more about the reasons why the vehicle is driving more efficiently.

5). **variance of steering angle and 6) variance of acceleration:** These metrics can be used to evaluate the improvement of the proposed algorithm on the stability of driving behavior. Simply increasing the number of lane-change may mean unstable behavior, and autonomous vehicle should improve the stability of driving behavior while ensuring flexibility to obtain efficient driving with concise maneuvers.

We compare our proposed methods and baselines in the following aspects:

- **Overall Performance:** Both in training and testing process, the overall performance represented by the 1) reward gained from RL agent. It can directly indicate the convergence of the RL algorithm and its approximate policy effectiveness.
- **Safety Performance:** This performance is reflected by the 2) collision rate metric. The lower the collision rate of the agent during training and testing, the safer the vehicle is driving. It is important to note that driving off the road boundary is also considered to be a collision.
- **Flexibility Performance:** An excessive focus on safety may lead to overly conservative driving policy. Autonomous vehicles need a certain degree of flexibility to adapt to complex scenarios, which is also a key reflection of the intelligence of autonomous driving algorithms. Flexibility can be measured by the 3) vehicle's average speed and 4) the number of lane changes. If the lane change frequency is high but the average speed remains low, it may indicate ineffective lane-change maneuvers. Moreover, these lane- change maneuvers are required to be safe. Reckless lane change does not imply flexibility. We expect EV to achieve faster speed with fewer lane changes, which requires EV to react flexibly and quickly. For instance, when the SV ahead is driving slowly and there is sufficient gap in the target lane for overtaking, EV should quickly complete the lane change maneuver to improve driving efficiency.
- **Stability Performance:** 5) variance of steering angle and 6) variance of acceleration are used to describe the stability of driving maneuvers. Greater granularity of the action space can bring about flexibility as well as instability in the action output. However, no one wants to see autonomous vehicle jerking or swaying from side to side on the road.

Training Results

The learning curves of total reward in training process are shown in Figure 4. All the agents were trained 5 times. Figure 4. shows their average reward curves and the corresponding variance distributions. It can be seen that

FIGURE 4 The total reward of different DRL agent in training process.

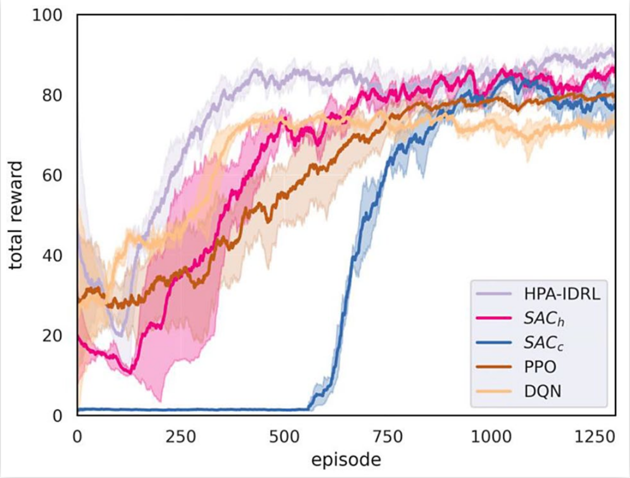
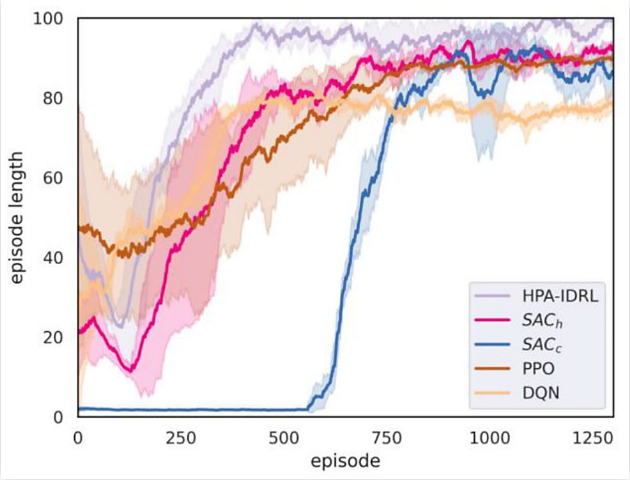


FIGURE 5 The episode length of different DRL agent in training process.

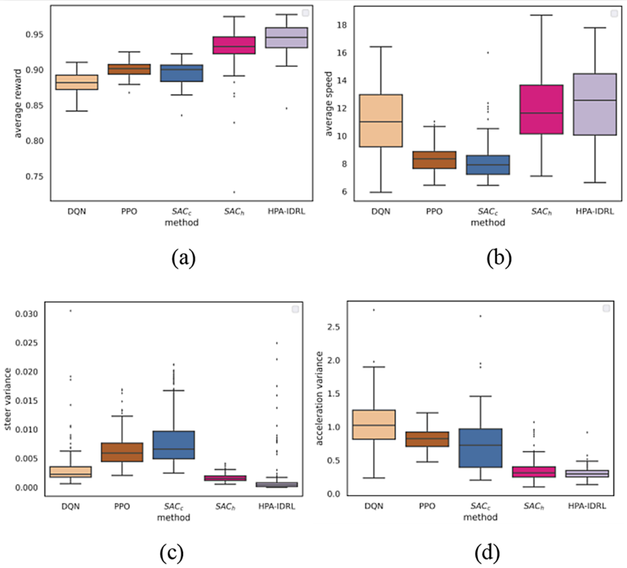


the proposed *HPA-IDRL* is able to achieve the highest reward convergence with a small fluctuation. *DQN*, *PPO* and SAC_h have similar results. Since *DQN* can only outputs long time-term discrete semantic actions, the distribution of the gained rewards are more concentrated during multi-times training. The convergence performance of *DQN* is the worst of all methods. *PPO*, SAC_c, and SAC_h have more variations in the rewards of the convergence process due to the greater granularity of the action space. Although this also brings a certain training instability, they all eventually converge to a slightly better performance relative to *DQN*. It is worth noting that SAC_c not gained reward in the early training stages due to the continuous control commands to directly control the vehicle. EV would frequently collide with SVs or road boundary. The policy network could not be updated until enough experiences have been collected in the experience replay batch. And then, SAC_c start learning in a right way to get higher rewards and eventually converge to a nice level as well as other agents. Figure 5. shows the learning curve of episode length, and their changing trends look similar to Figure 4. After convergence of, *HPA-IDRL* can reach almost full episodes length. which means that our proposed method can make the vehicle travel safely in the traffic until the end of the episode. This is one of the main reasons why it is able to harvest the most total reward. As a comparison, other methods still fall short in terms of episode length, which implies that collisions still often occur eventually.

Testing in Scenario 1

The first testing scenario is same as training (where the traffic density is set to 0.3), except for the random seed used in the SVs generation. Figure 6. illustrates the testing results of different DRL agent, including the average reward (Figure 6. (a)), average speed (Figure 6.(b)), the variance of steering angle (Figure 6.(c)) and of acceleration (Figure 6.(d)). These results show that the proposed method achieves the best performance in the test. Specifically, DQN has a low steer variance while a high acceleration variance. It means that the discrete semantic lateral behavior facilitates lateral driving stability, whereas it leads to longitudinal driving instability due to fewer choices of discrete acceleration actions. The average speed of *PPO* and SAC_c are really low, and they also have high steer variance. It suggests that their driving behavior is very unstable while driving inefficiently. SAC_c and *HPA-IDRL* utilize trajectory parameters that greatly enhance the stability of driving behavior. It is worth nothing that the proposed *HPA-IDRL* has the fastest average speed, which means the highest driving efficiency. It also has low variance of steering angle and acceleration, indicating EV achieves high-efficiency driving with smooth action.

FIGURE 6 The testing results of different DRL agent in scenario 1, (a) average reward, (b) average speed, (c) acceleration variance, (d) steer angle variance.



(a)

(b)

(c)

(d)

The quantitative statistical results are shown in Table 3, which demonstrates that, *HPA-IDRL* surpass all other agents. Taking *DQN* as a baseline algorithm, *PPO* and *SAC_c* agent can reduce the collision rate by approximately 88% and 96%. However, from the average speed and the average number of lane-change, it can be found that the reason for the low collision rate is that agents' maneuvering is too conservative. They lose about 25% and 27% of the average speed, while decreasing the number of lane changes by 43% and 77%. This may be because the actions of completely continuous control command are heavily influenced by the reward for tracking the reference path and the smoothness reward. As a result, the *PPO* and *SAC_c* agents tend to follow SVs at a very conservative speed. By generating trajectories, *SAC_h* and *HPA-IDRL* are able to significantly improve maneuvering flexibility while maintaining smoothness. Due to the lack of optimal consistency for discrete, continuous actions, the collision rate of *SAC_h* increases again to 10%, while the *HPA-IDRL* remains only 3%. Additionally, *HPA-IDRL* exhibits higher average speed and lower variance of both steering angle and acceleration. This is thanks in large part to the multi-critic with interaction reward and the interaction-aware attention mechanism. This enables the EV to better understand the surrounding traffic environment and output sample, smooth but efficient actions.

The results demonstrate that the proposed method effectively balances flexibility and smoothness, resulting in driving behavior that is both safe and efficient.

Testing in Scenario 2

In the scenario 2, the traffic density is increased to 0.5, which deviates a little from the training scenario. Therefore, the average speed of all agents is decreased and the collision rate is increased due to the more crowded traffic environment. Figure 7 and Table 4. Show the performance detail of different agents in this scenario. As in Scenario 1, *HPA-IDRL* achieves the best overall performance in all metrics. It achieves the highest average speed and lowest variance of control command while maintaining low collision rate. It is worth noting that the variance of steering angles and acceleration decreased for all agents. By observing the number of lane-change, it can be seen that due to the increase in traffic density, some lane-change maneuvers are replaced by car following maneuvers. Thus, in this situation, both speed and control fluctuations decrease accordingly. In summary, relatively unfamiliar test scenario tends to reduce the performance of all agents, but the proposed *HPA-IDRL* still maintains good safety, efficiency, and high driving behavior stability.

TABLE 3 Statistical quantitative results of different DRL agent in scenario 1.

metric					
agent	**DQN**	**PPO**	***SAC_c***	***SAC_h***	***HPA-IDRL***
average reward	0.88	0.90	0.89	0.93	**0.95**
collision rate (%)	0.51	0.06	0.02	0.10	**0.03**
average speed (m/s)	11.05	8.32	8.09	11.92	**12.35**
average lane-change number	9.77	5.61	2.22	6.98	**7.59**
steering angle variance (rad²)	32e-4	65e-4	85e-4	16e-4	**14e-4**
acceleration variance (m²/s²)	1.03	0.82	0.73	0.34	**0.30**

FIGURE 7 The testing results of different DRL agent in scenario 2, (a) average reward, (b) average speed, (c) acceleration variance, (d) steer angle variance.

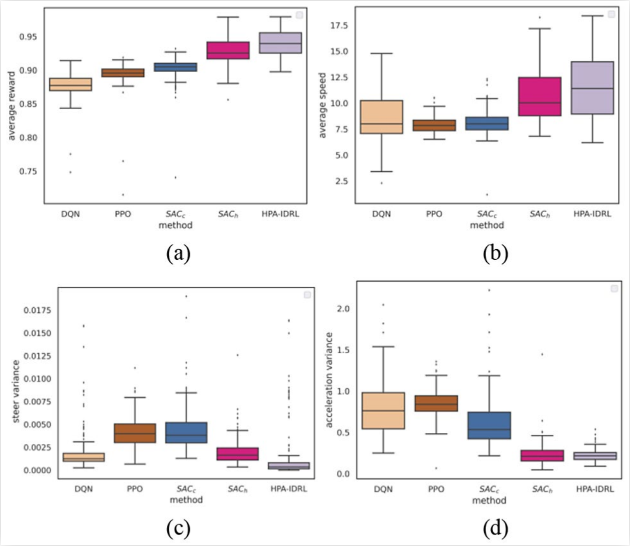


TABLE 4 Statistical quantitative results of different DRL agent in scenario 2.

metric					
agent	**DQN**	**PPO**	***SAC_c***	***SAC_h***	***HPA-IDRL***
average reward	0.87	0.89	0.90	0.92	**0.94**
collision rate (%)	0.56	0.09	0.04	0.12	**0.04**
average speed (m/s)	8.7	7.98	8.15	10.7	**11.45**
average lane-change number	7.62	4.99	2.20	6.31	**6.86**
steering angle variance (rad²)	19e-4	42e-4	45e-4	18e-4	**13e-4**
acceleration variance (m²/s²)	0.78	0.85	0.61	0.29	**0.22**

Conclusions

This paper proposes an interaction-aware reinforcement learning approach based on hybrid parameterized action space for autonomous driving in lane change scenario. The proposed method can output high-level feasible trajectory and low-level actuator control command to control the vehicle's motion together. Meanwhile, the reward functions for the local traffic environment are designed to evaluate the effect of the interaction between ego vehicle and surrounding vehicles. The contributions of the proposed method are summarized as: 1) propose a hybrid parameterized action based interaction-aware DRL framework (*HPA-IDRL*). It can output the parameterized action to generate feasible lane change path, which improves the stability of the lateral motion behavior while keeps the flexibility of the output by the DRL agent; 2) the proposed *HPA-IDRL* can learn from the reward not only considering self-benefits but also considering the benefits of the local traffic environment. Multiple critic networks are designed to evaluate Q values based on different reward functions, allowing RL agents to better focus on different driving object; 3) A multi-head attention layer is embedded before actor network and critic network respectively to exploit the interactive information in the traffic environment. Thus, the *HPA-IDRL* agent can generate more flexible and smooth driving behavior, which improves the safety and the efficiency of autonomous driving. The proposed method is implemented and validated with other four classic DRL agent in different simulation environments. The results show that the proposed *HPA-IDRL* can effectively balance the flexibility and smoothness of driving behavior, resulting in driving performance that is both safe and efficient.

References

1. Teng, S. et al., “Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives,” *IEEE Trans. Intell. Veh.* 8, no. 6 (2023): 3692-3711, doi:10.1109/TIV.20-23.3274536.
2. Li, Y., Deng, Z., Zeng, D. et al., “Lane-Change Planning with Dynamic Programming and Closed-Loop Forward Simulation for Autonomous Vehicle,” SAE Technical Paper 2021-01-7012, 2021, dol: 10.4271/2021-01-7012.
3. Feng, S., Yan, X., Sun, H., Feng, Y. et al., “Intelligent Driving Intelligence Test for Autonomous Vehicles with Naturalistic and Adversarial Environment,” *Nature Communication.* 12 (2021): 748, doi:10.1038/s41467-021-21007-8.
4. Ye, Z., “Baidu’s Mass Robotaxi Rollout Stirs Heated Debate in China,” Sixth Tone, July 12, 2024, https://www.sixth-tone.com/news/1015505.
5. Hang, P., Lv, C., Xing, Y., Huang, C. et al., “Human-Like Decision Making for Autonomous Driving: A Noncooperative Game Theoretic Approach,” *IEEE Trans. Intell. Transp. Syst.* 22, no. 4 (2021): 2076-2087, doi:10.1109/TITS.2020.3036984.
6. González, D., Pérez, J., Milanés, V., and Nashashibi, F., “A Review of Motion Planning Techniques for Automated Vehicles,” *IEEE Trans. Intell. Transp. Syst.* 17, no. 4 (2016): 1135-1145, doi:10.1109/TITS.2015.2498841.
7. Huang, Y., Wang, H., Khajepour, A. et al., “A Novel Local Motion Planning Framework for Autonomous Vehicles Based on Resistance Network and Model Predictive Control,” *IEEE Trans. Veh. Technol.* 69, no. 1 (2020): 55-66, doi:10.1109/TVT.20-19.2945934.
8. Claussmann, L., Revilloud, M., Gruyer, D., and Glaser, S., “A Review of Motion Planning for Highway Autonomous Driving,” *IEEE Trans. Intell. Transp. Syst.* 21, no. 5 (2020): 1826-1848, doi:10.1109/TITS.2019.2913998.
9. Xiong, G., Li, Y., Wang, S., Li, X. et al., “HMM and HSS Based Social Behavior of Intelligent Vehicles for Freeway Entrance Ramp,” *Int. J. Control Autom.* 7, no. 10 (2014): 79-90.
10. Li, N., Chen, H., Kolmanovsky, I., and Girard, A., “An Explicit Decision Tree Approach for Automated Driving,” in *Proceeding ASME Dynamic System and Control Conference*, 2017.
11. Li, Z., Hu, J., Leng, B., Xiong, L. et al., “An Integrated of Decision Making and Motion Planning Framework for Enhanced Oscillation-Free Capability,” *IEEE Trans. Intell. Transp. Syst.* 5, no. 6 (2024): 5718-5732, doi:10.1109/TITS.20-23.3332655.
12. Li, Z., Xiong, L., Leng, B., Fu, Z. et al., “Path Planning Method for Perpendicular Parking Based on Vehicle Kinematics Model Using MPC Optimization,” SAE Technical Paper 2022-01-0085 (2022), doi:10.4271/2022-01-0085.
13. Jayasree, K.R., Jayasree P.R., and Vivek, A., “Smoothed RRT Techniques for Trajectory Planning,” in *International Conference on Technological Advancements in Power and Energy (TAP Energy)*, 2017, doi: 10.1109/TAPENERGY.2017.8397376.
14. Huang, Y. et al., “A Motion Planning and Tracking Framework for Autonomous Vehicles Based on Artificial Potential Field Elaborated Resistance Network Approach,” *IEEE Trans. Ind. Electron.* 67, no. 2 (2020): 1376-1386, doi:10.1109/TIE.2019.28-98599.
15. Tu, C., Li, Z., Leng, B., and Xiong, L., “A Seamless Motion Planning Integrating Maneuver Decision Based on Hybrid Model Predictive Control,” *Proc IEEE Intell. Transp. Syst. Conf. (ITSC)* (2023), doi:10.1109/ITSC57777.2023.10422155.
16. Ly, A.O. and Akhloufi, M., “Learning to Drive by Imitation: An Overview of Deep Behavior Cloning Methods,” *IEEE Trans. Intell. Veh.* 6, no. 2 (2021): 195-209, doi:10.1109/TIV.2020.3002505.
17. Yuan, K. et al., “Evolutionary Decision-Making and Planning for Autonomous Driving: A Hybrid Augmented Intelligence Framework,” *IEEE Trans. Intell. Transp. Syst.* 25, no. 7 (2024): 7339-7351, doi:10.1109/TITS.2023.3349198.
18. Kaufmann, E., Bauersfeld, L., et.al. “Champion-Level Drone Racing Using Deep Reinforcement Learning”, *Nature*, 620: 982-987, 2023, doi: 10.1038/s41586-023-06419-4.
19. Aradi, S., “Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles,” *IEEE Trans. Intell. Transp. Syst.* 23, no. 2 (2022): 740-759, doi:10.1109/TITS.2020.3024655.
20. Ye, Y., Zhang, X., and Sun, J., “Automated Vehicle’s Behavior Decision Making Using Deep Reinforcement Learning and High-Fidelity Simulation Environment,” *Transp. Res. Part C, Emerg. Technol.* 107 (2019): 155-170, doi:10.1016/j.trc.2019.08.011.

21. Zhang, Y., Gao, B., Guo, L., Guo, H. et al., "Adaptive Decision-Making for Automated Vehicles Under Roundabout Scenarios Using Optimization Embedded Reinforcement Learning," *IEEE Trans. Neural Netw. Learn. Syst.* 32, no. 12 (2021): 5526-6638, doi:10.1109/TNNLS.2020.3042981.
22. Kiran, B.R., Sobh, I., Talpaert, V. et al., "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Trans. Intell. Transp. Syst.* 23, no. 6 (2022): 4909-4926, doi:10.1109/TITS.2021.3054625.
23. Li, Z., Xiong, L., Leng, B., Xu, P. et al., "Safe Reinforcement Learning of Lane Change Decision Making with Risk-Fused Constraint," *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)* (2023), doi:10.1109/ITSC57777.2023.10422331.
24. Lu, X., Fan F.X., and Wang, T., "Action and Trajectory Planning for Urban Autonomous Driving with Hierarchical Reinforcement Learning," arXiv: 2306.15968, 2023.
25. Zheng, H., Chen, C., Li, S. et al., "Learning-Based Safe Control for Robot and Autonomous Vehicle Using Efficient Safety Certificate," *IEEE Open J. Intell. Transp. Syst.* 4 (2023): 419-430, doi:10.1109/OJITS.2023.3280573.
26. Cheng, R., Orosz, G., Murray, R.M., and Burdick, J.W., "End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks," *Proc. AAAI Conf. Artif. Intell.* 33, no. 1 (2019): 3387-3395, doi:10.1609/aaai.v33i01.33013387.
27. Mysore, S., Cheng, G., Zhao, Y., Saenko, K. et al., "Multi-Critic Actor Learning: Teaching RL Policies to Act with Style," in *Proceeding International Conference Learning Represent (ICLR)*, 2022.
28. Wang, Y., Wang, L., Guo, J. et al., "Ego-Efficient Lane Changes of Connected and Automated Vehicles with Impacts on Traffic Flow," *Transp. Res. Part C, Emerg. Technol.* 138 (2022): 103478, doi:10.1016/j.trc.2021.103478.
29. Wang, Z., Zhang, S., Feng, X., and Sui, Y., "Autonomous Underwater Vehicle Path Planning Based on Actor-Multi-Critic Reinforcement Learning," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.* 235, no. 10 (2021): 1787-1796, doi:10.1177/09596-51820937085.
30. Vaswani, A., Shazeer, N., Parmar, N. et al., "Attention Is All You Need," *Adv. Neural Inf. Proces. Syst. (NeurIPS)* 30 (2017): 5998-6008.
31. Leurent, E., "An Environment for Autonomous Driving Decision Making," GitHub Repository, 2018, https://github.com/eleu-rent/highway-env.
32. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. et al., "Proximal Policy Optimization Algorithms," arXiv: 1707.06347, 2017.
33. Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S., "Soft Actor-Critic: Offpolicy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceeding International Conference Mechanic Learning (PMLR)*, 1861-1870, 2018.
34. Yang, R., Li, Z., Leng, B., and Xiong, L., "Safe Reinforcement Learning for Autonomous Vehicles to Make Lane-Change Decisions: Constraint Based on Incomplete Information Game Theory," in *CAA International Conference on Vehicular Control and Intelligence*, 2023, doi: 10.1109/CVCI59596.2023.10397427.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant 52372394, in part by the National Key Research and Development Program of China under Grant 2022YFE0117100, in part by the Science and Technology Commission of Shanghai under Grant 21DZ1203802, and in part by Shanghai Automotive Industry Science and Technology Development Foundation under Grant 2203.

Contact Information

Zhuoren Li
School of Automotive Studies
Tongji University
Shanghai 201804, China
1911055@tongji.edu.cn

Bo Leng
School of Automotive Studies
Tongji University
Shanghai 201804, China
lengbo@tongji.edu.cn

Definitions/Abbreviations

RL - Reinforcement Learning

DRL - Deep Reinforcement Learning

DNN - Deep Neural Network

EV - Ego Vehicle

SVs - Surrounding Vehicles

DQN - Deep Q-Network

PPO - Proximal Policy Optimization

SAC - Soft Actor Critic

Positions and opinions advanced in this work are those of the author(s) and not necessarily those of SAE International. Responsibility for the content of the work lies solely with the author(s).

This paper is based upon a presentation at the *2024 Intelligent and Connected Vehicles Symposium*.

ISSN 0148-7191