2024 IEEE 27th International Conference on
Intelligent Transportation Systems (ITSC)
September 24- 27, 2024. Edmonton, Canada

Stability Enhanced Hierarchical Reinforcement Learning for Autonomous Driving with Parameterized Trajectory Action

Guizhe Jin[1], Zhuoren Li[1], Bo Leng[1], Wei Han[1], Lu Xiong[1], Jia Hu[2] and Nan Li[1]

2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3315-0592-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITSC58415.2024.10920092

Abstract— **Reinforcement Learning (RL) has become a potential method for autonomous driving to adapt to complex driving environments with high flexibility. However, the popular RL paradigm directly outputting the vehicle control commands makes the future motion with fluctuation. To improve the driving behavior stability of RL method while ensuring the motion flexibility, this paper proposes a stability enhanced hierarchical reinforcement learning method based on parameterized trajectory action (RL-PTA). It offers feasible driving path in the long horizon and real-time control commands in the short horizon simultaneously. The RL agent actively contributes to path generation with discrete-continuous hybrid parameter actions, and the parameterized action space also ensures optimal consistency of the hybrid output. The experiment results show that the proposed method can generate flexible and stable lane-change driving behavior, thereby improving the efficiency and safety for autonomous driving.**

I. Introduction

Decision-making and control directly affect the safety and flexibility of autonomous driving, which is considered the brain of autonomous driving [1], [2]. At the same time, manually designed rule-based systems find it challenging to adapt to complex driving environments. As a very successful method in the field of sequential decision problems [3], [4], reinforcement learning (RL) has become a highly promising learning paradigm for autonomous driving, especially in decision-making and control tasks [5-7]. However, the popular RL paradigm that directly generates short-term control commands (such as steering angle and acceleration) results in future motion characterized by large uncertainties, which results in deficiencies in the smoothness and stability of driving behavior [8]. This is also one of the reasons hindering its large-scale practical applications.

On one hand, when RL agent outputs are short-term control commands, the RL policy network typically fits the probability distribution of control commands under different observed state inputs. In this context, the output commands are easy to change frequently with the time step [9], [10]. On other hand, due to the lack of explicit long-term action planning, the RL agent insufficiently considers the feasibility of future motion, and the future trajectory of the vehicle is still unknown [10]. Hence, the real-time RL control commands are prone to sudden changes in dynamic environments.

In lane-change scenarios with structured roads, the goal of lateral motion is usually explicit and remains unchanged for a long time, which is suitable to be represented by discrete semantic behaviors in long horizon [11]. Therefore, some studies have designed discrete path sets based on the discrete lane-change semantic behaviors, enabling the RL agent to directly choose from them [11-13]. The lane-change trajectory generated by these methods is finite, however, a long-term lane-change goal can be achieved through an infinite number of feasible trajectories. Thus, these trajectory selection methods restrict the maneuverability of the vehicle's motion. Some researchers attempted to enable RL agent to generate continuously variable target points with discrete semantic behaviors for trajectory planning [14], [15]. It promotes both the stability and maneuverability of lane-change driving behavior to some extent through the trajectory planner. Nevertheless, the longitudinal speed control is also planned by the rule-based planner according to the target points, which still loses some flexibility of the RL agent. Additionally, these methods often directly discretize continuous output actions to generate the discrete semantic behavior, which may lose the advantages of continuous action space for fine-grained control [16].

To this end, this paper simultaneously considers the long-term discrete lane-change behavior goal and short-term real-time vehicle motion control. Based on **P**arameterized **T**rajectory **A**ctions, a hierarchical **R**einforcement **L**earning method with a hybrid action output is designed to enhance the stability of driving behavior in lane-change scenarios while keeping the flexibility, thereby improving the driving efficiency and safety. The proposed method is called **RL-PTA** and the contributions of this paper are summarized as:

- Proposes a stability enhanced hierarchical reinforcement learning framework to achieve smooth and flexible driving behavior in dynamic traffic environments;
- Enables the RL agent to participate in path generation with the parameterized trajectory action and thus to adapt to various scenes;
- Realizes hybrid action output based on parameterized action space. It can synchronously generate lane-change trajectory targets over the long horizon and real-time acceleration control commands over the short horizon, hence the discrete and continuous actions have optimal

⋆This work was supported by the National Key R&D Program of China under Grant No. 2021YFB2501201 and No. 2022YFE0117100, the National Natural Science Foundation of China under Grant 52372394, the Science and Technology Commission of Shanghai under Grant No. 21DZ1203802 and Fundamental Research Funds for the Central Universities. *(corresponding author: Bo Leng.)*

[1]Guizhe Jin, Zhuoren Li, Bo Leng, Wei Han, Lu Xiong and Nan Li are with the School of Automotive Studies, Tongji University, Shanghai 201804, China. (email: jgz13573016892@163.com; 1911055@tongji.edu.cn; leng_bo@tongji.edu.cn; tjhanwei@foxmail.com; xiong_lu@tongji.edu.cn; li_nan@tongji.edu.cn)

[2]Jia Hu is with the College of Transportation Engineering, Tongji University, Shanghai 201804, China. (email: hujia@tongji.edu.cn;)

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.

consistency.

The remainder of this paper is organized as follows: Section II presents the system framework of the proposed RL-PTA for lane-change maneuvers. In Section III, we introduce the details of the methodology. The analysis of the experiment results based on the simulation environment and the real traffic dataset is presented in Section IV. The conclusion of the paper is in Section V.

II. System Framework

A. Preliminary

The RL agent learns the driving policy by interacting with the environment through exploitation and exploration with the trial-and-error paradigm [17]. The sequential decision and control problem can be formed as a Markov Decision Process (MDP) represented by $< S,A,P,R,\gamma >$ [13], where S is the state space, A is the action space, P is the transition probability, R is the reward function, and γ is the discount factor. RL agent selects an action $a_t \in A$ at each time step t according to the current state $s_t \in S$. It receives a numerical reward R_{t+1} and transitions to a new state s_{t+1}. The sequence $\{s_0,a_0,R_1,s_1,a_1,R_2,...\}$ is called a rollout [18]. The policy $\pi(a|s)$ is defined as the probability of selecting action a at state s. The cumulative discounted reward starting from time-step t is defined as $G_t = \sum_{j\geq t} \gamma^{j-t} R(s_j,a_j)$. Hence, the action-value function of policy π can be defined as the following:

$$Q^{\pi}(s_t,a_t) = \mathbb{E}\left[G_t \,|s_t = s, a_t = a\right]. \tag{1}$$

The goal of RL agent is to find the optimal policy that maximizes the expected reward. This optimal action-value function can be described by the Bellman equation,

$$Q(s_t,a_t) = \mathbb{E}\left[R_t + \gamma \max_{a' \in A} Q(s_{t+1},a') | s_t = s, a_t = a\right]. \tag{2}$$

B. MDP with Parameterized Action Space

To represent the MDP in a trajectory parameterized action space, the action needs to be defined within a hierarchical architecture. Firstly, the high-level discrete action k is selected from a set of discrete actions K. After that, the low-level parameter action $x_k \in X_k$ is chosen corresponding to k. Here, X_k represents a continuous set for $\forall k \in K$. The low-level continuous parameter is optional, and different discrete actions can share common low-level parameters [19]. Therefore, the discrete-continuous hybrid action space can be expressed as,

$$A = \{(k,x_k) \mid x_k \in X_k, \forall k \in K\}. \tag{3}$$

For $\forall a \in A$, $s \in S$, the action-value function $Q(s,a) = Q(s,k,x_k)$. Let k_t represent the discrete action at time t and let x_{k_t} be the associated continuous parameter. Then Eq. (2) can be rewritten as,

$$Q(s_t,k_t,x_{k_t}) = \mathop{\mathbb{E}}_{R_t,s_{t+1}}\left[R_t + \gamma \max_{k\in K} \sup_{x_k \in X_k} Q(s_{t+1},k,x_k) | s_t = s, a_t = (k_t,x_{k_t})\right]. \tag{4}$$

The deep neural network (DNN) $Q(s,k,x_k;\omega)$ is employed to approximate $Q(s,k,x_k)$, where ω represents the network parameters. Additionally, a deterministic policy network $\mu(s;\theta)$ is utilized to approximate the determination of x_k, where θ denotes the parameters for the policy network. Taking the extremes in the continuous space X_k is computationally intractable. However, when the value function is given, x_k is a function of state s for any state s and the high-level action k. On this basis, the network can be utilized to solve the problem of taking extremes in a continuous space. Further, it can be understood as the process of exploring θ when ω is given:

$$Q(s,k,\mu(s;\theta);\omega) \approx \sup_{x_k \in X_k} Q(s,k,x_k;\omega), \forall k \in K. \tag{5}$$

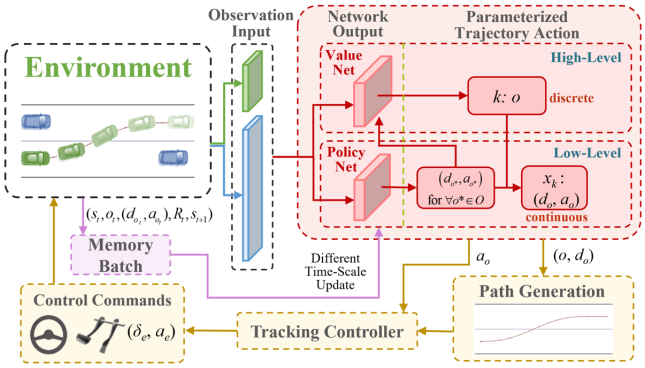

Fig. 1. The overall framework of proposed RL-PTA. The o^* is a temporary parameter until final o is selected, representing all possible discrete actions.

This paper uses the high-level trajectory objective $o \in O$ to describe the lateral lane-change long-term behaviors in the structured roads, where O is a set of feasible discrete semantic behaviors. The o represents the overall driving intent, preliminarily guaranteeing safety and flexibility. Meanwhile, the low-level trajectory objective and acceleration command are denoted by continuous actions $(d_o,a_o) \in (D_o,A_o)$, where (D_o,A_o) are two continuous action sets for $\forall o \in O$. Specifically, d_o and a_o have a higher correlation with the specific future state of the vehicle, which further ensures flexibility and safety while improving the smoothness of vehicle motion. Thus, the optimal action-value function for the lane-change task can be expressed as,

$$Q(s_t,o_t,(d_{o_t},a_{o_t})) = \mathop{\mathbb{E}}_{R_t,s_{t+1}}\left[R_t + \gamma \max_{o\in O}\{\sup_{D_o,A_o} Q(s_{t+1},o,(d_o,a_o))\} | s_t = s\right]. \tag{6}$$

The parameterized action space enables discrete and continuous actions to be optimized within a unified framework to achieve higher value functions, where the optimization principle of both actions are consistent. The high-level action can be effectively achieved through the low-level action, while the low-level action can be flexibly adjusted to adapt to the changes in the high-level action, and ultimately, which makes the hybrid action have optimal consistency.

C. Overall Framework of RL-PTA

Fig. 1 illustrates the overall system framework of the proposed RL-PTA. The observation information of the ego-

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.

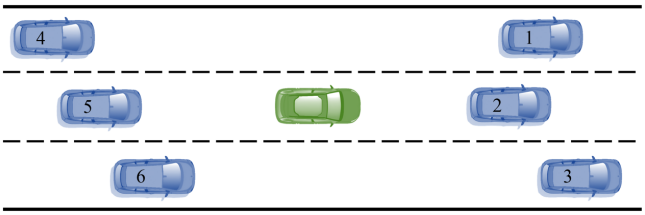

Fig. 2. Orientation of six SVs in relation to EV.

vehicle (EV) and surrounding vehicles (SV) from the environment is processed and input into deep neural networks which have parameterized action outputs. By the hybrid hierarchical RL network including value network and policy network, high-level discrete semantic action o and low-level continuous action (d_o, a_o) are simultaneously obtained, providing the acceleration command and the objective of generating a feasible lateral path. Through a designed tracking controller, the steering angle δ_e and acceleration a_e could be implemented to EV. The driving experiences in the memory batch are used to update the RL network with different time scales according to Eq. (6).

III. Methodology

A. RL Agent Formulation

1) State Space Definition: The state space is vectorized by constructing the information vector s^e of the EV and s^s of the SV.

For s^e, it can be expressed as:

$$s^e = [ID_{lane}, d_{lat}, v_{lat}, v_{lon}, a_{lat}, a_{lon}], \tag{7}$$

where ID_{lane} denotes the current lane id of EV, and d_{lat} denotes the lateral distance of EV with respect to the current lane. Meanwhile, v_{lat}/v_{lon} and a_{lat}/a_{lon} are the lateral/longitudinal velocity and acceleration, respectively.

Considering six SVs around EV, the illustration of their position relative to EV is shown in Fig. 2. For s^s, it can be expressed as:

$$s^s = \left[\Delta d_{lat_i}, \Delta d_{lon_i}, \Delta v_{lat_i}, \Delta v_{lon_i}, \Delta a_{lat_i}, \Delta a_{lon_i}\right], \tag{8}$$

where $i = 1 \sim 6$, d_{lat} and d_{lon} respectively denote the lateral and longitudinal distances of the SV with respect to EV. Further, $\Delta v_{lat}/\Delta v_{lon}$ and $\Delta a_{lat}/\Delta a_{lon}$ denote the lateral/longitudinal velocity and acceleration of the SV relative to EV. Thus, the total state vector can be represented as $s = (s^e, s^s)$.

2) Parameterized Action-Space: The parameterized action space is a hybrid space containing the discrete high-level lane-change objective as well as the continuous low-level path target and acceleration command. It thereby provides a motion planning foundation for implementing complex decisions. Thus, the action space can be specified as:

$$A = \{o, (d_o, a_o) \mid (d_o, a_o) \in (A_o, D_o), \forall o \in O\}. \tag{9}$$

where o is the discrete high-level lane-change objective selected from $O = \{LCL: 1, LK: 0, LCR: -1\}$, which means lane change left, lane keeping, and lane change right. The $d_o \in D_o$ is the continuous low-level path target, which serves as a parameter for constructing the target path, and D_o varies according to the EV's current speed and vehicle kinematic model:

$$D_o \sim \left[\min\left(\sqrt{4R_0 l_c - l_c^2}, \frac{v_{lon}^2}{2a_{\max}^-}\right), e^{k_v \cdot |v_{lon}| + l_c}\right], \tag{10}$$

where R_0 represents the minimum turning radius of the vehicle, l_c denotes the lane width, $a_{\max}^-$ signifies the maximum braking acceleration of the vehicle, and speed weight $k_v = 1$. The $a_o \in A_o$ is the EV's acceleration command, and A_o is a continuous set of $[-3\text{m/s}^2, 3\text{m/s}^2]$.

3) Reward Design: The design of the reward dictates the driving strategy adopted by the EV. Consequently, the reward function primarily includes efficiency reward R_e, safety reward R_s, and smoothness reward R_c:

$$R = R_e + R_s + R_c. \tag{11}$$

For the efficiency reward R_e, it receives a higher positive reward when EV's speed is closer to the target speed v_t. Meanwhile, a negative reward for too low speed is also necessary to prevent extremely slow driving or even stopping. Therefore, R_e can be expressed by the following equation:

$$R_e = k_{e1} \cdot \frac{|v - v_t|}{v_t} - k_{e2} \cdot \max(0, \frac{v_l - v}{v_l}), \tag{12}$$

where v_l is the low-speed threshold for the negative reward, and k_{e1} and k_{e2} are weights for the two efficiency sub-rewards, set to $k_{e1} = k_{e2} = 1$.

To ensure the safety reward R_s, it is crucial to prevent collisions between the EV and SVs, assigning a significant negative reward when a collision occurs. Additionally, a positive reward associated with the Time-to-Collision (TTC) is introduced. Thus, R_s can be denoted by the following equation:

$$R_s = -k_{s1} f_{\text{coll}} + k_{s2} \cdot \text{sat}_{[0,1]}\left(\frac{\Delta t}{t_{\max}}\right), \tag{13}$$

where f_{coll} is a flag bit of collision, which is set to 1 when a collision occurs or EV is off the road, otherwise to 0. The TTC of EV and its front vehicle is represented by Δt, and $t_{\max}$ is the upper limit for evaluating the TTC. The weights k_{s1} and k_{s2} are set to $k_{s1} = 10$, and $k_{s2} = 0.5$.

For the smoothness reward R_c, the EV should be encouraged to maintain moderate the steering angle δ_e and the acceleration a_e while driving. Therefore, R_c can be denoted by:

$$R_c = -k_{c1} \cdot \frac{|\delta_e|}{|\delta_{\max}|} - k_{c2} \cdot \frac{|a_e|}{|a_{\max}|}, \tag{14}$$

where $\delta_{\max}$ and $a_{\max}$ represent the maximum steering angle and the maximum acceleration that the EV can safely handle on the road. The weights k_{c1} and k_{c2} are set to $k_{c1} = k_{c2} = 0.5$.

B. Path Generation and Tracking

Considering path continuity and computational efficiency, we employ a fifth-degree polynomial curve for path generation based on the parameters (o, d_o), which is represented by:

$$y^j = \sum_{j=0}^{5} c_j x^j. \tag{15}$$

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.

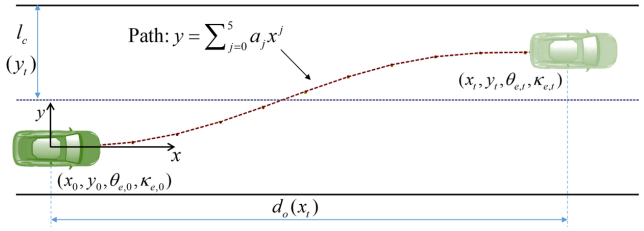

Fig. 3. An example of lane-change for EV.

As shown in Fig. 3, the EV's position state in the road coordinate system is $(x, y, \theta_e, \kappa_e)$, where θ_e is the yaw angle and κ_e is the curvature. Consequently, $(x_0, y_0, \theta_{e,0}, \kappa_{e,0}) = (0, 0, \theta_{e,0}, \kappa_{e,0})$ denotes the initial position state of EV, while $(x_t, y_t, \theta_{e,t}, \kappa_{e,t}) = (d_o, o \cdot l_c, 0, 0)$ denotes the position state of the target path point. According to the discrete high-level lane-change objective o and continuous low-level path target d_o provided by the RL-agent, coefficients of the fifth degree polynomial path for EV can be calculated by the previous work [13]. It fully capitalizes on the high flexibility of the RL agent to generate future motion paths that adapt to dynamic scenes.

C. Network Design

For parameterized action space A defined in Eq. (9), discrete and continuous actions are respectively selected from the action-value networks $Q(s, o, (d_o, a_o); \omega)$ and the policy network $\mu(s; \theta)$. Thus, Eq. (6) is written as following form:

$$Q(s_t, o_t, (d_{o_t}, a_{o_t})) = \mathop{\mathbb{E}}_{R_t, s_{t+1}} \left[R_t + \gamma \max_{o \in O} Q(s_{t+1}, o, \mu(s_{t+1}; \theta); \omega) \right]. \quad (16)$$

For the parameter ω, it is estimated by minimizing the mean square Bellman error through gradient descent. Additionally, constructing the target action-value network $Q'(s, o, (d_o, a_o); \omega')$ and the target policy network $\mu'(s; \theta')$, where ω' and θ' are progressively closer to ω and θ based on soft update technique. Specifically, at step t, the target y_t can be defined as:

$$y_t = R_t + \gamma \max_{o \in O} Q'\left(s_{t+1}, o, \mu'\left(s_{t+1}; \theta'\right); \omega'\right). \quad (17)$$

Based on Eq. (16) and Eq. (17), the loss function l_t for updating the Q-network and the μ-network can be expressed as:

$$l_t^Q(\omega) = \frac{1}{2}[y_t - Q(s_t, o_t, (d_{o_t}, a_{o_t}); \omega)]^2, \quad (18)$$

$$l_t^\mu(\theta) = -\sum_{o \in O} Q(s_t, o, \mu(s_t; \theta); \omega). \quad (19)$$

Furthermore, the gradient during network update can be represented as follows:

$$\nabla l_t^Q(\omega) = [Q(s_t, o_t, (d_{o_t}, a_{o_t}; \omega) - y_t)] \nabla_\omega Q, \quad (20)$$

$$\nabla l_t^\mu(\theta) = -\sum_{o \in O} \nabla_\theta \mu(s_t; \theta) \nabla_{\mu(s_t;\theta)} Q(s_t, o, \mu(s_t; \theta)). \quad (21)$$

In the ideal case, the learning goal is to minimize $l_t^\mu(\theta)$ with a given ω_t. It is worth noting that the Q-network of high-level long-term lane-change objective should be updated on a long-time scale, while the μ-network of low-level continuous action requires a short-time scale. Therefore, online approximation using a two-time scale update rule is employed in this work. The step size α_t for updating ω is asymptotically negligible compared to the step size β_t for updating θ. The training process of the proposed RL-PTA is explicitly introduced in Alg. 1.

Algorithm 1 Training process of proposed RL-PTA

Input: Step sizes $\{\alpha_t, \beta_t\}$,total training steps N, exploration parameter ε, learning rate l_r, soft-update parameter τ.

Initialize: experience memory batch D, networks $\{Q, \mu, Q', \mu'\}$ with random parameters $\{\omega, \theta, \omega', \theta'\}$.
for $t = 0$ to N **do**
  Get state s_t from environment.
  Select $(d_{o_t}, a_{o_t}) = \mu(s_t; \theta)$.
  Select $o_t = \begin{cases} \text{random choice, with } \varepsilon \\ \max_{o \in O} Q(s_t, o, (d_{o_t}, a_{o_t})), \text{with } 1-\varepsilon \end{cases}$
  Create motion path by o_t and d_{o_t} .
  Get δ_t based on path, together with a_{o_t} to control EV.
  Get s_{t+1} and R_t from environment.
  Store transition $\{s_t, (o_t, d_{o_t}, a_{o_t}), R_t, s_{t+1})\}$ into D.
  Sample randomly from D to compute $l_t^Q(\omega), l_t^\mu(\theta)$.
  Update $\omega_{t+1} \leftarrow \omega_t - \alpha_t \nabla_\omega l_t^Q(\omega)$.
  Update $\theta_{t+1} \leftarrow \theta_t - \beta_t \nabla_\theta l_t^\mu(\theta)$.
  $\omega'_{t+1} \leftarrow \tau\omega_t + (1-\tau)\omega_t^{'}$, $\theta'_{t+1} \leftarrow \tau\theta_t + (1-\tau)\theta_t^{'}$.
  $s_t \leftarrow s_{t+1}$.
  if s_t is terminal **then**
    Reset environment.
  end if
end for
return

IV. Experiment Results Analysis

A. Implementation Setting

1) Simulation Environment Setup: For the training and testing of all methods, we constructed typical structured road simulation scenarios using the TAD Sim 2.0 platform. The three-lane scenario is created where EV is randomly generated in the center line of one lane. SVs are also randomly generated, and their longitudinal and lateral driving behaviors use the IDM/MOBIL model [20], [21]. The desired speed of EV is set slightly higher than the average speed of SVs to encourage lane changes. Additionally, SVs execute their maneuvers with a PID controller and some of them might change lanes at the proper time to get closer to the desired speed. The simulation step size is set to 0.2s.

Meanwhile, we introduce the concept of traffic capacity to accurately describe the distribution of vehicles, typically denoted by V/C [22]. The V/C ratio for all simulated scenarios in this paper is set to 0.5, indicating a moderately congested traffic environment.

2) Algorithm Parameters Setup: To ensure the fairness in training and testing, all algorithms share the same hyperparameters, as shown in Table I. In addition, the ADAM [23] algorithm is also used to optimize the network to improve computationally efficient.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.

TABLE I

SHARED HYPER-PARAMETERS

Item	Value
Number of hidden layers	3
Hidden layer size	256
Activation function	Leaky_ReLU
Discount factor γ	0.9
Learning rate for value and policy	1e-4
soft-update parameter τ	0.005
Replay buffer size	40000
Sample batch size	256
Number of steps for training N	150000
Number of episodes for testing	200
Maximum number of steps per episode	100

3) Dataset for Validation: To justify the superiority of our method in real scenarios, this paper presents a validation based on the HighD dataset [24], which is a large-scale natural vehicle trajectory dataset from German highways. The dataset contains all the information necessary for validation, such as the lateral and longitudinal positions of each vehicle, speed, acceleration, lane position, and so on. Specifically, we select cars with lane-change behavior as EVs and let their behavior be controlled by the DRL agent. For the SVs, they still follow the established trajectory. Meanwhile, to match with the training process, the speeds of all SVs are scaled to below 20 m/s. And the maximum time for each episode during validation is 40s.

B. Compared Methods

To analyze the effectiveness and sophistication of our method, we select DQN, SAC with continuous action (SAC_con), and SAC with hybrid action (SAC_hybrid) as baseline methods to compare training and test results with the proposed RL-PTA. To ensure fair comparisons, all four methods are tested under identical simulation environment settings and the same random seed.

1) DQN: It outputs both lateral and longitudinal high-level semantic behavior actions. Then, a PID controller generates the steering angle and acceleration to control the EV's motion.

2) SAC_con: The policy network outputs the steering angle and acceleration to directly control the EV.

3) SAC_hybrid: It has the same action outputs $\{o, x_o, a_o\}$ as the RL-PTA and is able to generate motion trajectory. The difference is that this baseline directly discretizes part of its continuous output to obtain o.

C. Simulation Results Analysis

All methods are trained for 150,000 time steps with varying episode sizes, and all of them converge to the optimal driving strategy after 1300 episodes. Fig. 4 shows the evolution of total reward, average speed and episode length for each episode in training process. DQN and SAC_hybrid agents eventually learn similar average driving speeds, but the latter can better avoid collisions to get longer episode length and reward. DQN has limited attention on safety, resulting in the poorest performance in total reward and episode length. For SAC_con, the random exploration of continuous control commands often leads to off-road excursions

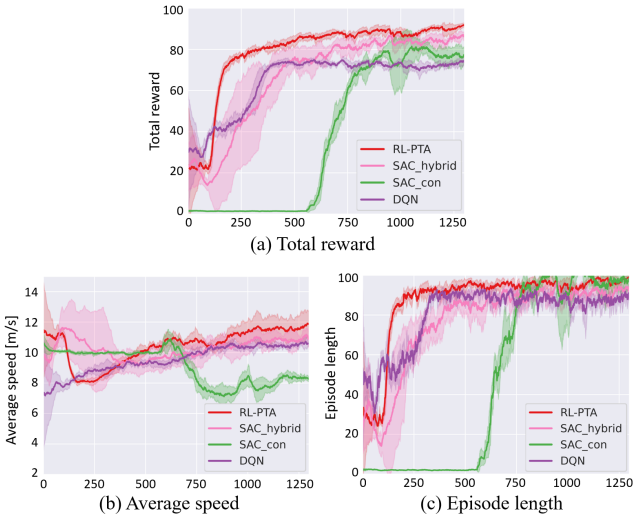

Fig. 4. Training results of each method.

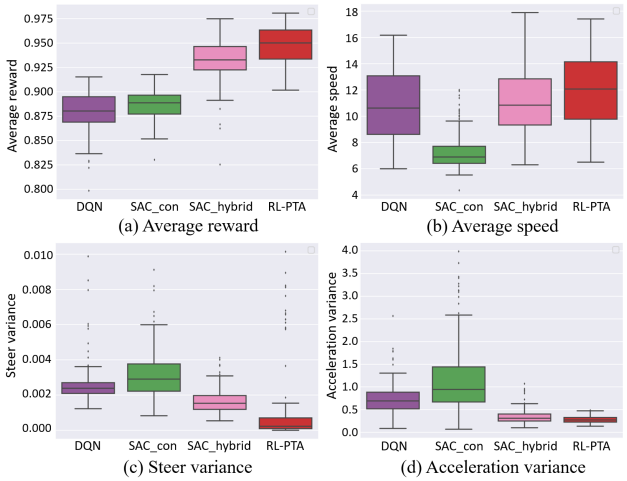

Fig. 5. Distribution of some key indicators while testing in the simulation environment.

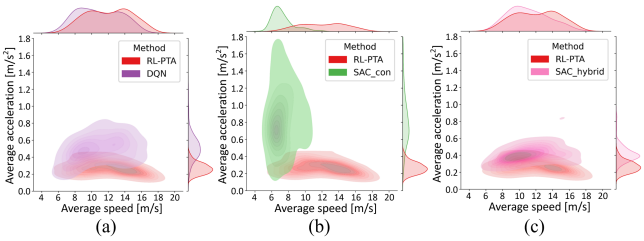

Fig. 6. Distribution comparison of speed and acceleration while testing in the simulation environment.

in early training process, which contribute very short episode length and low reward. And SAC_con eventually learns a very conservative behavior, which is reflected in its very low average speed. Obviously, the proposed RL-PTA achieves higher reward and faster convergence than other methods, as shown in Fig. 4(a). More importantly, the RL-PTA agent quickly learns flexible driving behaviors from experience, leading to improvements in both average speed and episode

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.

TABLE II
KEY INDICATORS OF TEST RESULTS IN THE SIMULATION ENVIRONMENT

method	average reward	average speed	episode length	lane-change times	collision rate	steering variance	acc variance
DQN	0.880	10.9	89.5	10.22	0.40%	0.0026	0.728
SAC_con	0.887	7.3	98.9	2.09	0.01%	0.0031	1.159
SAC_hybrid	0.923	11.1	91.4	7.36	0.11%	0.0015	0.344
RL-PTA	**0.948**	**12.1**	**95.2**	**8.21**	**0.04%**	**0.0009**	**0.278**

length. Therefore, our method demonstrably learns higher-quality driving behavior, enhancing both flexibility and stability.

After training process, we test each method with 200 episodes. The distributions of some key indicators, which reflect the details of driving behavior, are shown in Fig. 5. From Fig. 5(a), it can be seen that RL-PTA is still able to obtain higher reward. In Fig. 5(b), the average speed per episode of RL-PTA is generally higher, which effectively ensures the driving efficiency of EV. Fig. 5(c) and Fig. 5(d) show the distributions of the steering variance and acceleration variance, respectively, where RL-PTA maintains the lowest values for both, demonstrating its ability to enhance the stability of driving behavior.

The more quantitative details of the key indicators for the four methods are shown in Table II. Besides, Fig. 6 shows the joint distribution of velocity and acceleration, comparing RL-PTA with each baseline method separately. It can be found that RL-PTA achieves the best performance in almost all metrics, while SAC_hybrid is second to it. This proves that the parameterized action space can effectively improve the driving performance for RL agent. Specifically, comparing to SAC_hybrid, RL-PTA increases the average reward by 2.7%. Additionally, it also increases the average speed and the number of lane-change times per episode by 9.0% and 11.5%, respectively, while the collision rate drops by 63.6%. These results suggest that RL-PTA can driving more effectively and safely, making more lane changes when necessary while minimizing the risk of collisions. Furthermore, RL-PTA exhibits respectively 40.0% and 19.2% lower steering and acceleration variance, compared to SAC_hybrid, indicating its superior ability to maintain lateral and longitudinal stability during driving. In Fig. 6(c), the joint distribution of speed and acceleration for RL-PTA is concentrated towards the lower right, indicating its ability to achieve higher speeds with smaller acceleration fluctuations. These results demonstrate the superiority of the parameterized action space for handling hybrid actions compared to directly discretizing partially continuous actions.

It is worth pointing out that DQN has the maximum number of lane-change times, but its collision rate is much higher than other methods. This implies that utilizing only the DRL agent to output high-level behavior goals will limit flexibility and make driving strategy more aggressive, hence more dangerous. In contrast, allowing the DRL agent to be more directly involved in EV's control, such as RL-PTA, can significantly ameliorate this problem and improve flexibility. For SAC_con, it always tends to choose a more conservative

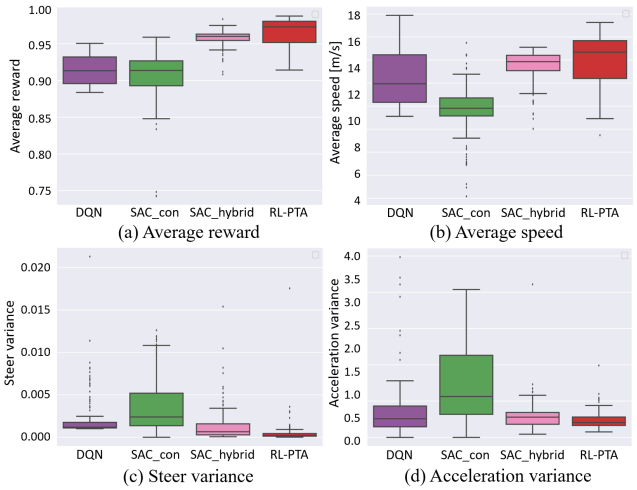

Fig. 7. Distribution of some key indicators while validating with HighD dataset.

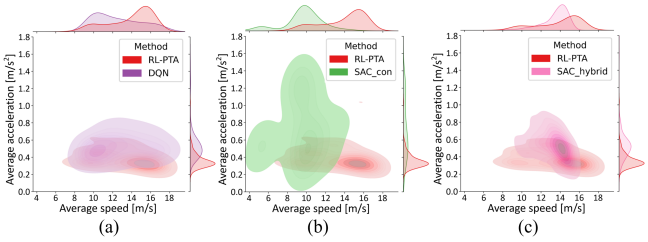

Fig. 8. Distribution comparison of speed and acceleration while validating with HighD dataset.

driving strategy to maximize long-term reward, which results in having the lowest collision rate but at the cost of significant sacrifice of efficiency. As shown in Fig. 6(b), SAC_con also has high acceleration variance and steering variance, indicating that it maintains unstable maneuvers even the speed is low. In contrast, RL-PTA's parameterized trajectory action enables both flexible and stable driving, effectively balancing efficiency and safety.

D. Validation in HighD Dataset

To further validate the effectiveness of RL-PTA, all methods are tested in HighD dataset and Fig. 7. shows the distribution of several key indicators. Obviously, RL-PTA still maintains the highest speed and reward, while achieving the lowest steering variance and acceleration variance. It demonstrates the effectiveness of our method in real traffic scenarios, which can enhance the driving efficiency as well as the stability the of driving behavior. As shown in Table III,

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.

TABLE III
KEY INDICATORS OF VALIDATION RESULTS WITH HIGHD DATASET

method	average reward	average speed	episode length	lane-change times	collision rate	steering variance	acc variance
DQN	0.914	13.1	87.2	7.99	0.80%	0.0021	0.340
SAC_con	0.904	9.5	99.7	1.69	0.00%	0.0037	0.737
SAC_hybrid	0.958	14.3	91.8	5.54	0.11%	0.0012	0.292
RL-PTA	**0.965**	**14.9**	**95.0**	**6.32**	**0.05%**	**0.0006**	**0.235**

the collision rate of RL-PTA remains nearly unchanged compared to that in Table II, and it is still lower than that of DQN and SAC_hybrid. This demonstrates the robustness of our method in adapting to different lane-change scenes and ensuring vehicle safety. Further, Fig. 8(c) shows that RL-PTA still maintains superior acceleration control in the HighD dataset, further highlighting the effectiveness of the parameterized trajectory action.

E. Discussion

In summary, the training and testing results highlight the superiority of our RL-PTA over the other three baseline methods. Additionally, we validated the effectiveness of RL-PTA using the HighD dataset. The parameterized trajectory action enables more flexible and smoother lane-change behavior, enhancing the efficiency and safety of EV's motion. Moreover, the experimental results demonstrate that the parameterized action space significantly improves the stability of hybrid action output compared to directly discretizing partially continuous actions.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a hierarchical reinforcement learning method based on parameterized trajectory action (RL-PTA), generating both long-term feasible driving paths and short-term real-time control commands simultaneously. The parameterized action space is instrumental in ensuring optimal consistency between the discrete and continuous components of the hybrid actions, leading to enhanced the stability of driving behaviors. Experimental results demonstrate that RL-PTA enables autonomous vehicle to perform more flexible and smoother lane changes, improving both driving efficiency and safety in structured road scenarios. Future work will focus on further enhancing safety while maintaining the efficiency of vehicle operation, and will also explore the applicability of RL-PTA in more complex traffic scenarios.

REFERENCES

[1] H. Deng, Y. Zhao, Q. Wang and A.T. Nguyen, "Deep Reinforcement Learning Based Decision-Making Strategy of Autonomous Vehicle in Highway Uncertain Driving Environments," *Automot. Innov.* vol.6, pp. 438-452, 2023.

[2] Z. Li, J. Hu, B. Leng, L. Xiong and Z. Fu, "An Integrated of Decision Making and Motion Planning Framework for Enhanced Oscillation-Free Capability," *IEEE Trans. Intell. Transp. Syst.*, early access, 2023, doi: 10.1109/TITS.2023.3332655.

[3] V. Mnih et al., "Human-level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] N. Brown and T. Sandholm. "Superhuman AI for Heads-up No-limit Poker: Libratus Beats Top Professionals," *Science*, vol. 359, no. 6374, pp. 418– 424, 2018.

[5] B. R. Kiran, I. Sobh, V. Talpaert, et al, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[6] L. Chen et al., "Milestones in Autonomous Driving and Intelligent Vehicles: Survey of Surveys," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 2, pp. 1046-1056, Feb. 2023.

[7] X. He and C. Lv, "Towards Safe Autonomous Driving: Decision Making with Observation-Robust Reinforcement Learning," *Automot. Innov.* vol.6, pp. 509-520, 2023.

[8] S. R. Jaladi, Z. Chen, N. R. Malayanur, R. M. Macherla and B. Li, "End-To-End Training and Testing Gamification Framework to Learn Human Highway Driving," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 4296-4301, 2022.

[9] X. Tang, B. Huang, T. Liu and X. Lin, "Highway Decision-Making and Motion Planning for Autonomous Driving via Soft Actor-Critic," *IEEE Trans. Veh. Tech.*, vol. 71, no. 5, pp. 4706-4717, May. 2022.

[10] Q. Liu, X. Li, S. Yuan and Z. Li, "Decision-Making Technology for Autonomous Vehicles: Learning-Based Methods, Applications and Future Outlook," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 30-37, 2021.

[11] C. -J. Hoel, K. Wolff and L. Laine, "Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 2148-2155, 2018.

[12] K. B. Naveed, Z. Qiao and J. M. Dolan, "Trajectory Planning for Autonomous Vehicles Using Hierarchical Reinforcement Learning," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 601-606, 2021.

[13] Z. Li, L. Xiong, B. Leng, P. Xu and Z. Fu, "Safe Reinforcement Learning of Lane Change Decision Making with Risk-Fused Constraint," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 1313-1319, 2023.

[14] X. Lu, F.X. Fan and T. Wang, "Action and Trajectory Planning for Urban Autonomous Driving with Hierarchical Reinforcement Learning." *arXiv preprint*, arXiv: 2306.15968, 2023.

[15] Z. Gu et al., "Safe-State Enhancement Method for Autonomous Driving via Direct Hierarchical Reinforcement Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9966-9983, Sep. 2023.

[16] Z. Fan, R. Su, W. Zhang and Y. Yu, "Hybrid Actor-Critic Reinforcement Learning in Parameterized Action Space," *arXiv preprint*, arXiv:1903.01344, 2019.

[17] S. B. Thrun, "Efficient Exploration in Reinforcement Learning," *Carnegie Mellon University*, 1992.

[18] Z. Zhu and H. Zhao, "A Survey of Deep Rl and Il for Autonomous Driving Policy Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14043-14065, 2021.

[19] Xiong, Jiechao, et al. "Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space." *arXiv preprint*, arXiv:1810.06394, 2018.

[20] M. Treiber, A.Hennecke and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical review E*, vol. 62, no. 2, 1805, 2000.

[21] A. Kesting, T. Martin and H. Dirk, "General Lane-changing mMdel MOBIL for Car-Following Models," *Transportation Research Record*, vol. 1999, no.1, pp. 86-94, 2007.

[22] R. Baldacci, P. Toth, and D. Vigo, "Exact Algorithms for Routing Problems Under Vehicle Capacity Constraints," *Annals of Operations Research*, vol. 175, pp. 213-245, 2010.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein, "The Highd Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," *IEEE Trans. Intell. Transp. Syst.*, pp. 2118–2125, 2018.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on March 24,2025 at 05:21:54 UTC from IEEE Xplore. Restrictions apply.