Applications of Distributional Soft Actor-Critic in Real-world Autonomous Driving

1st Jingliang Duan
School of Vehicle and Mobility
Tsinghua University
Beijing, 100084, China
email: duanjl15@163.com

2nd Fawang Zhang
School of Mechanical Engineering
Beijing Institute of Technology
Beijing, 100084, China
email: Fawang_Troy_Zhang@163.com

3rd Shengbo Eben Li*
School of Vehicle and Mobility
Tsinghua University
Beijing, 100084, China
email: lishbo@tsinghua.edu.cn

4th Yangang Ren
School of Vehicle and Mobility
Tsinghua University
Beijing, 100084, China
email: ryg18@mails.tsinghua.edu.cn

5th Bo Cheng
School of Vehicle and Mobility
Tsinghua University
Beijing, 100084, China
email: chengbo@tsinghua.edu.cn

6th Zhe Xin
College of Engineering
China Agricultural University
Beijing, 100084, China
email: xinzhe@cau.edu.cn

2022 2nd International Conference on Computer, Control and Robotics (ICCCR) | 978-1-6654-6674-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCCR54399.2022.9790288

Abstract—**Reinforcement learning (RL) plays an important role in the decision-making of high-level autonomous vehicles due to the self-evolving ability without reliance on labeled data. Although existing RL-based decision-making studies have yielded fruitful results, most of them are carried out based on simulation platforms. Due to the inherent difference between simulation and the real world, it is of great significance to verify the efficacy of RL-based decision-making in practical applications. In this paper, a multi-lane driving task and the corresponding reward function are designed to provide a basis for RL-based policy learning. The distributional soft actor-critic algorithm is used to learn an offline policy based on a simulated environment. Then, we implement the learned policy to a real car on a two-lane park road. Both objective and subjective experiments are carried out to verify the effectiveness and robustness of the learned policy in practical applications. Experimental results show the trained policy can not only complete driving tasks smoothly and robustly, but also acquire fair satisfaction from subjects. Our work provides certain evidence for the feasibility of RL in real-world driving tasks.**

Index Terms—**DSAC, decision-making, autonomous driving.**

I. Introduction

Recently, significant progress has been made by reinforcement learning (RL) in many challenging domains, from MOBA games to robot manipulation [1]. Compared with decision-making methods based on rules or supervised learning, RL enables agents to constantly learn control policies by interacting with the environment through trial and error. Thereby, as for autonomous driving, RL is also a promising remedy to realize the self-evolving of self-driving decision-making ability without relying on rules and labeled driving data.

Before the advent of deep RL, Ngai *et al.* realized lane changing and overtaking in a curved multi-lane simulation environment using the celebrated Q-Learning algorithm [2]. All states and actions, such as heading angle and steering angle, were discretized, and the corresponding Q value of each state-action is recorded based on tables. With the development of deep RL, RL-based autonomous driving has been widely studied. Lillicap *et al.* proposed the first deep RL algorithm suitable for continuous controller settings, called DDPG, and realized lane-keeping function with the simulated image as input on the TORCS driving simulation platform [3]. Subsequently, similar driving functions have been achieved using a variety of mainstream RL algorithms, such as DDPG [4], [5], A3C [6], inverse RL [7], and DQN [8]. In 2018, Wayve corporation applied the DDPG algorithm to the real vehicle equipped with a monocular camera and obtained a fairly good lane-following effect on rural country roads [9].

The aforementioned studies directly learn a driving policy based on raw sensor data, such as images. Isele *et al.* showed that compared with raw sensor information, the perceived indicators, such as relative speed and distance, will greatly reduce the dimension of state space and improve the performance of the learned policy [10]. Under this scheme, Duan *et al.* learned a vector-based driving policy using parallel RL, which can successfully realize car-following, lane-changing, and overtaking on a two-lane simulated highway with dense traffic [11]. The driving scenario is described by a 26-dimensional state vector, considering 4 nearest surrounding vehicles as well as the road information. Guan *et al.* employed the PPO algorithm to develop a cooperation control for 8 connected automated vehicles at an unsignalized simulated intersection, where each vehicle is represented by its velocity and distance to the center of the intersection [12].

Although existing RL-based decision-making studies have yielded fruitful results, most of them are carried out based on simulated environments, such as TORCS and CARLA. Due to the inevitable difference between simulation and

This research work is supported by NSF China with 52072213, and U20A20334. It is also partially supported by Tsinghua University-Didi Joint Research Center for Future Mobility. J. Duan and F. Zhang contributed equally to this work. All correspondence should be sent to S. E. Li.

J. Duan is also with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore.

F. Zhang is also with the College of Engineering, China Agricultural University, Beijing, 100084, China.

978-1-6654-6674-5/22/$31.00 ©2022 IEEE

actual applications, it is of great significance to verify the effectiveness of the learned policy on a real vehicle. In this paper, a multi-lane driving task and the corresponding reward function are designed to provide the basis for RL-based policy learning. The distributional soft actor-critic (DSAC) algorithm [13] is used to learn an offline policy based on a simulated environment. Then, we implement the learned policy to a real car on a two-lane park road. Both objective and subjective experiments are carried out to verify the effectiveness and robustness of the learned policy in practical applications.

The rest of the paper is organized as follows: Section II describes the distributional soft actor-critic algorithm. Section III describes the driving task, state, and reward function. Section IV shows the experimental results. Section V concludes this paper.

II. Preliminaries

A. Basic Descriptions of RL

We model the sequential decision-making problem for self-driving as a Markov decision process (MDP). A typical MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the reward function, $p : S \times A \times S \rightarrow \mathbb{R}$ is the transition probability distribution, and $\gamma \in (0, 1)$ is the discount factor. For each state s, the action is chosen according to a stochastic policy $\pi(a|s)$: $\mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$. At current state s_t, the vehicle will take action a_t according to current policy π and then the environment will return next state s_{t+1} according to the environment model $p(s_{t+1}|s_t, a_t)$, i.e, $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$ and a scalar reward r_t. The current and next state-action pairs are also denoted as (s, a) and (s', a'). This process will repeat until one episode is finished. We will slightly abuse notation by denoting the state or state-action distribution induced by π as ρ_π.

The maximum entropy RL [14], [15] aims to maximize the expected accumulated reward and policy entropy, by augmenting the standard RL objective with an entropy maximization term

$$\begin{aligned} J_\pi &= \mathop{\mathbb{E}}_{(s_i,a_i)\sim\rho_\pi} \Big[\sum_{i=t}^{\infty} \gamma^{i-t}[r(s_i, a_i) - \alpha \log \pi(a_i|s_i)]\Big] \\ &= \mathop{\mathbb{E}}_{s\sim\rho_\pi, a\sim\pi} [Q^\pi(s, a) - \alpha \log \pi(a|s)]. \end{aligned} \tag{1}$$

where $-\log\pi(a|s)$ is the entropy of current policy to increase exploration ability during the learning process, α is the coefficient that determines the relative importance of the entropy term against the reward, and the Q-value of policy π is defined as

$$Q^\pi(s_t, a_t) = r_t + \mathop{\mathbb{E}}_{\substack{s_{i>t}\sim p,\\ a_{i>t}\sim\pi}} [\sum_{i=t+1}^{\infty} \gamma^{i-t}[r_i - \alpha \log \pi(a_i|s_i)]]. \tag{2}$$

B. Distributional soft actor-critic (DSAC)

We proposed the distributional soft actor-critic (DSAC) algorithm under the framework of maximum entropy RL earlier, which is one of the SOTA RL algorithms for continuous control tasks [13]. In the sequel, we will utilize this algorithm to learn the driving policy. Unlike the mainstream RL algorithms, where a single value function $Q(s, a)$ is directly optimized, DSAC attempts to learn the distribution of return to reduce Q-value overestimation. First, we define the random state-action return as

$$Z^\pi(s_t, a_t) = r(s_t, a_t) + \sum_{i=t+1}^{\infty} \gamma^{i-t}[r(s_i, a_i) - \alpha\log\pi(a_i|s_i)],$$

where $s_i \sim p$ and $a_i \sim \pi$. Obviously, the expectation of the return satisfies

$$Q^\pi(s, a) = \mathop{\mathbb{E}}_{a_i\sim\pi} [Z^\pi(s, a)].$$

Based on the definition of $Z^\pi(s_t, a_t)$, the distributional Bellman operator can be derived as

$$\mathcal{T}^\pi Z^\pi(s, a) \overset{D}{=} r(s, a) + \gamma(Z^\pi(s', a') - \log \pi(a'|s'))\big|_{s'\sim p, a'\sim\pi},$$

where $A \overset{D}{=} B$ denotes that two random variables A and B have equal probability laws. Denote the distribution of $Z^\pi(s, a)$ as $\mathcal{Z}^\pi(s, a)$, i.e., $Z^\pi(s, a) \sim \mathcal{Z}^\pi(s, a)$. For practical applications, the return distribution can be updated by

$$\mathcal{Z}_{\text{new}} = \arg\min_{\mathcal{Z}} \mathop{\mathbb{E}}_{a\sim\pi} \big[D_{\text{KL}}(\mathcal{T}^\pi \mathcal{Z}_{\text{old}}(\cdot|s, a), \mathcal{Z}(\cdot|s, a))\big], \tag{3}$$

where D_{KL} represents Kullback-Leibler (KL) divergence and $\mathcal{T}^\pi Z_{\text{old}}(\cdot|s, a) \sim \mathcal{T}^\pi \mathcal{Z}_{\text{old}}(\cdot|s, a)$.

Since Q-value is the expected value of the return distribution, the policy can be directly updated by

$$\pi_{\text{new}} = \arg\max_{\pi} \mathop{\mathbb{E}}_{a\sim\pi} \big[Q^\pi(s, a) - \alpha \log \pi(a|s)\big]. \tag{4}$$

It has been proved that policy evaluation step in (3) and policy improvement step in (4) can gradually converge to the optimal policy [13].

III. Problem Design

A. Driving Task Description

To verify the effectiveness of the DSAC algorithm in actual autonomous vehicles, we design an autonomous driving experiment based on a two-lane park road. The experiment road is shown in Fig. 1, which has a total length of 170m, with two speed bumps. Each lane is 3.5m wide. The experimental vehicle is a Chang-an CS55 SUV (see Fig. 2) equipped with a decision industrial PC (IPC) to send control commands and a simulated IPC to simulate virtual traffic flow. The specific model of the decision IPC is KMDA3211. The driving indicators of the ego vehicle, such as position and heading angle, can be measured by Real-time kinematic (RTK) systems and will be sent to the control IPC via serial communication. The receiving and sending of the steering wheel angle are completed through the CAN bus. The 51Sim-One driving simulator is installed on the simulated IPC to generate random surrounding traffic, whose information will be sent to the control IPC through ethernet communication.

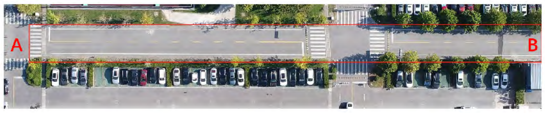


Fig. 1. Two-lane park road.

Fig. 2. Experimental car.

The ego vehicle aims to drive from side A to side B without colliding with surrounding vehicles. In the designed driving task, the ego vehicle only focuses on lateral decision-making and control, while the vehicle speed is controlled by a speed tracking system, with 20km/h as the expected speed. To implement the DSAC algorithm in this driving task, we first use DSAC to learn a driving policy offline based on a driving simulator. Then, the learned policy will be embedded into the controller IPC to make online decisions according to the indicators collected from RTK systems and the simulated IPC.

B. State and Action

To learn a effective driving policy using RL, we first need to design the state, which is composed of information about surrounding vehicles, the ego vehicle, and road. For each surrounding vehicle, we consider 6 indicators, including relative longitudinal and lateral distances (D_{long} and D_{lat}), speed v_{other}, the relative heading angle to lane centerline Φ_{other}, the vehicle length and width (L and W). The information of the ego vehicle comprises the longitudinal speed v_{ego}, lateral speed v_y, yaw rate Υ, heading angle Φ_{ego}, steering wheel angle ξ, longitudinal acceleration acc_x, lateral acceleration acc_y. Besides, the state also consists of the road geometry information, including lateral deviation relative to lane center D_{center}, distance to the left and right road edge (D_{left} and D_{right}).

As for the decision action, we choose the increment of the steering wheel angle, denoted as $\Delta\xi \in [-\frac{\pi}{9}, \frac{\pi}{9}]$, to realize lateral control and ensure the continuity of steering angle. The target steering wheel angle can be calculated as $\xi_{\text{exp}} = \xi + \Delta\xi$.

C. Reward Function

To rationally guide policy learning, we design a reward function by considering driving compliance, safety, and smoothness, expressed as

$$r = \begin{cases} R_{\text{bad}}, & \text{failure} \\ R_{\text{smooth}} + R_{\text{legal}} + R_{\text{safe}}, & \text{else} \end{cases}, \tag{5}$$

where $R_{\text{bad}} = -5000$ is a large negative reward to punish some devastating events, such as collisions or driving off the road.

R_{smooth} aims to make a more comfortable driving process by penalizing control variables $\Delta\xi$ and certain states of the ego vehicle:

$$R_{\text{regular}} = -100{\xi_{\text{exp}}}^2 - 200\Delta\xi^2 - 200{\Phi_{\text{ego}}}^2 - 100\Upsilon^2.$$

Compliance reward R_{legal} assures the ego vehicle to ride inside the road:

$$R_{\text{legal}} = -\,10{D_{\text{center}}}^2 - 40(1 - \tanh(4\min\{D_{\text{left}}, D_{\text{right}}\})).$$

Finally, R_{safe} considers the lateral and longitudinal gap between the ego vehicle and each surrounding vehicle, denoted as D_{LatGap}, D_{LongGap} respectively, to punish the decreasing trend of relative distance:

$$\begin{aligned} R_{\text{safe}} = 30 - {}& \\ 40 \sum_{veh \in \mathcal{V}} & \text{sgn}(-D_{\text{LatGap}})\text{sgn}(D_{\text{long}})\Big(1 - \tanh\frac{D_{\text{LongGap}}}{v_{\text{ego}}}\Big) \\ - 20 \sum_{veh \in \mathcal{V}} & \text{sgn}(-D_{\text{LatGap}})\text{sgn}(-D_{\text{long}})\Big(1 - \tanh\frac{D_{\text{LongGap}}}{v_{\text{other}}}\Big) \\ - 40 \sum_{veh \in \mathcal{V}} & \text{sgn}(-D_{\text{LongGap}})\Big(1 - \tanh(1.5D_{\text{LatGap}})\Big), \end{aligned}$$

where $\text{sgn}(\cdot)$ means the sign function and $\mathcal{V}$ is the set of surrounding vehicles.

D. Learning Details

We built a simulated version of the driving environment in Fig. 1 using SUMO [16]. Then, the DSAC algorithm is employed to learn a driving policy offline (see [13] for more algorithm details). The representation method presented in [17] is used to describe surrounding vehicles. Both distributional value function and policy are approximated by multi-layer perceptron (MLP) with 5 hidden layers and 128 units per layer. The hyperparameters used in training are listed in Table I and the training results are shown in Fig. 3. The average return converges after about 0.6 million iterations.

TABLE I: Training hyperparameters

Name	Value
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Activation	GELU
Batch size	256
Discounted factor(γ)	0.99
Policy learning rate	$5 \times 10^{-5} \rightarrow 4 \times 10^{-5}$
Value learning rate	$8 \times 10^{-5} \rightarrow 4 \times 10^{-5}$
Learning rate of α	$1 \times 10^{-4} \rightarrow 4 \times 10^{-5}$
Target entropy($\overline{\mathcal{H}}$)	$\overline{\mathcal{H}} = -2$

IV. Experimental Results

In this section, we implement the learned policy on a real vehicle to conduct objective and subjective experiments. Experimental results verify the efficacy of the driving policy learned by DSAC.

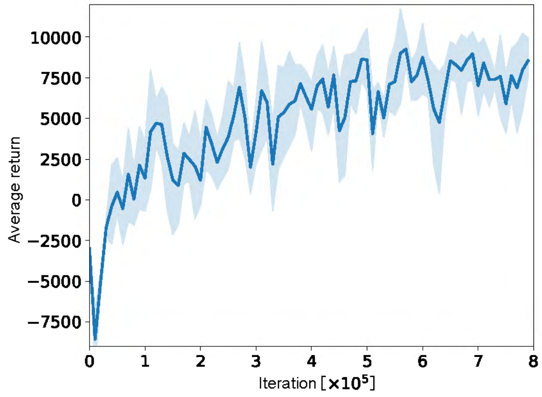


Fig. 3. Learning curve. The solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over 5 runs.

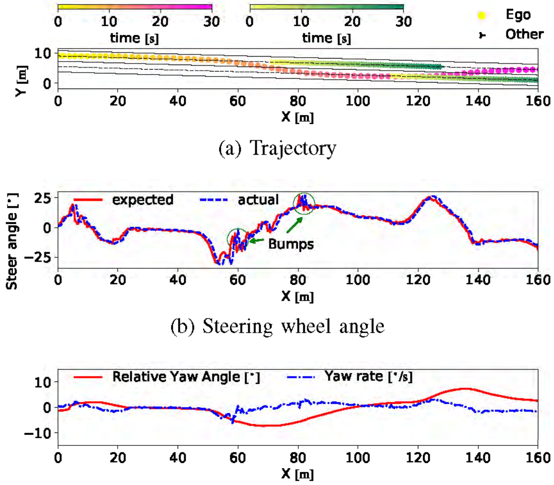
A. Objective Experiments

To perform more precise validation, we set a random initialization for the ego vehicle. As mentioned in Section III-A, the speed of the ego vehicle is fixed at 20km/h, and we only need to control the lateral motion by sending the expected steering wheel angle. Surrounding vehicles provided by simulated IPC, are set to ride at a random speed between 0 and 20km/h with random lane-changing behavior.

Fig. 4 shows a typical autonomous driving process with two surrounding vehicles. We can see from Fig. 4a that the ego vehicle first chooses a right lane change due to the slow speed of the front car; after that, it gradually approaches another front car in the right lane. Therefore, the ego vehicle changes to the left lane again and rushes towards the road end. From Fig. 4b, the actual steering wheel angle changes smoothly during the entire driving process, and only small oscillations occur when the vehicle passes through two bumps. Besides, we can see that there exists about 100ms delay between the expected angle and the actual one due to the mechanical response. Similarly, the yaw rate and relative heading angle maintain a smooth trend during riding in Fig. 4c, which indicates that our algorithm can assure a satisfying level of driving comfort.

Fig. 5 displays the decision-making trajectories corresponding to different initial positions and surrounding traffic. Experimental results show that the policy successfully learns useful maneuvers such as lane-keeping, lane-changing, and overtaking, so as to smoothly realize autonomous driving in response to different surrounding vehicles.

To assess the sensitivity of the learned policy to state noise, we applied 6 different levels of sensing noise to certain states before being observed by the ego car during the experiment. We assume that the sensing noise obeys the zero-mean Gaussian distribution. The standard deviations of the additional noise under different levels are shown in Table II. Fig. 6 displays the impact of noise level on certain driving states, such as lateral derivation and steering wheel angle, while driving straight. Results show that the variance of these states is proportionally correlated to the noise level. However, even at the maximum noise level, the lateral deviation is still less than 20cm. This indicates that the learned policy has high robustness to perceptual noise.



(a) Trajectory

(b) Steering wheel angle

(c) Heading angle and yaw rate

Fig. 4. Visualization of trajectory and states.

TABLE II: The standard deviation of sensor noise

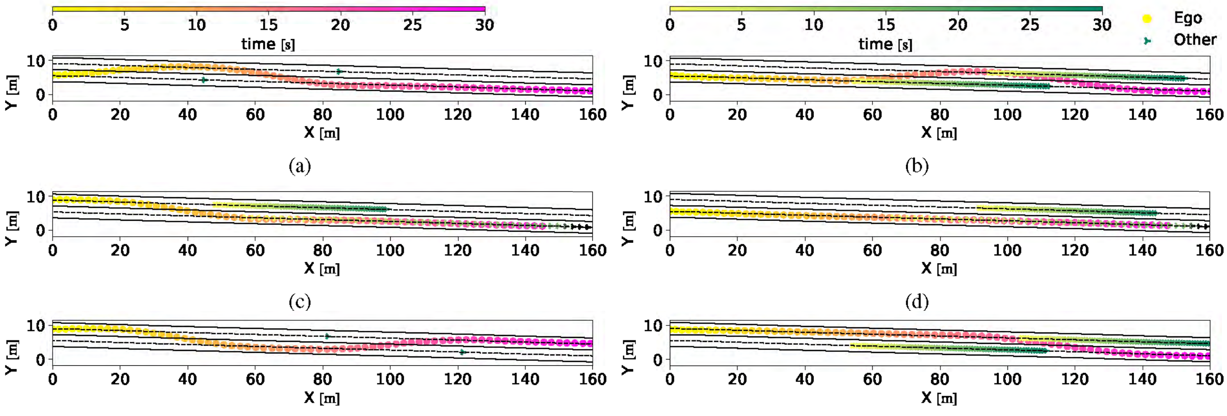
Noise level	1	2	3	4	5	6
D_{center} [m]	0.017	0.033	0.05	0.066	0.083	0.10
Φ_{ego} [°]	0.17	0.33	0.50	0.66	0.83	1.00
D_{long}/ D_{lat} [m]	0.05	0.10	0.15	0.20	0.25	0.30
v_{other} [m/s]	0.05	0.10	0.15	0.20	0.25	0.30
Φ_{other} [°]	1.4	2.8	4.2	5.6	7	8.4
L/W [m]	0.01	0.02	0.03	0.04	0.05	0.06

[1] Each data in the table represents the standard deviations of additional noise applied to different observed indicators.

B. Subjective Experiments

Besides the objective experiments, we also conducted a subjective evaluation, in which 20 subjects (including 9 women) were recruited to ride in the vehicle and describe their feelings about certain driving functions. These participants were aged between 25 and 45, and 18 of them held legal driving licenses with driving experience ranging from 1 to 14 years. Each subjective was asked to take two trial rides. As shown in Fig. 7, two monitors were installed to display the surrounding traffic flow. After experiments, subjects were required to fill out a questionnaire anonymously, which contains 5 problems about the lane-changing and lane-keeping functions, driving comfort, and dependability (see Table III for details).

The final results are shown in Fig. 8. From Fig. 8a, we can see 85% of subjects believe the learned driving policy can match or outperform human drivers in terms of lane-keeping operation. As for lane change, all participants gave a non-negative point in Fig. 8b, and 80% of them believe the lane-changing timing is appropriate in Fig. 8d, indicating that our system can realize reasonable lane-changing and overtaking. In addition, according to Fig. 8c, only one subjective held the view that the level of driving comfort has decreased. From Fig. 8e, regarding the trust in our system , 40% of subjects



(a)

(b)

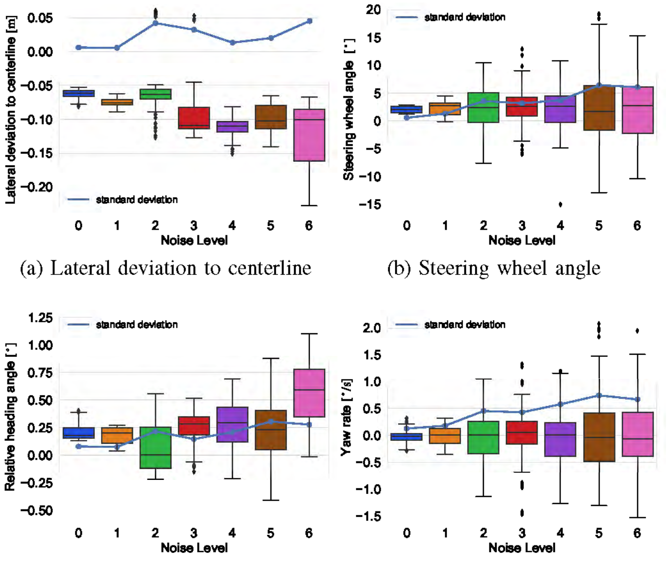
(c)

(d)

(e)

(f)

Fig. 5. Experimental results under different traffic conditions.



(a) Lateral deviation to centerline

(b) Steering wheel angle

(c) Relative heading angle

(d) Yaw rate

Fig. 6. The impact of noise on driving states. Each boxplot is drawn based on 100 values.

TABLE III: Subjective evaluation questionnaire

Survey questions
Scoring criteria [-2,2]: -2∼much worse, 0∼similar, 2∼much better Q1. Compared with human drivers, how good is the lane-keeping operation of the learned policy? Q2. Compared with human drivers, how good is the lane-changing operation of the learned policy? Q3. Compared with human drivers, how good is the driving comfort of the learned policy?
Scoring criteria [-2,2]: -2∼premature, 0∼suitable, 2∼too late Q4. What do you think of the timing of the lane change?
Scoring criteria [-2,2]: -2∼distrustful, 0∼wait-and-see, 2∼trustful Q5. Do you trust the decision system?

(a)

(b)

Fig. 7. Illustration of subjective experiment.

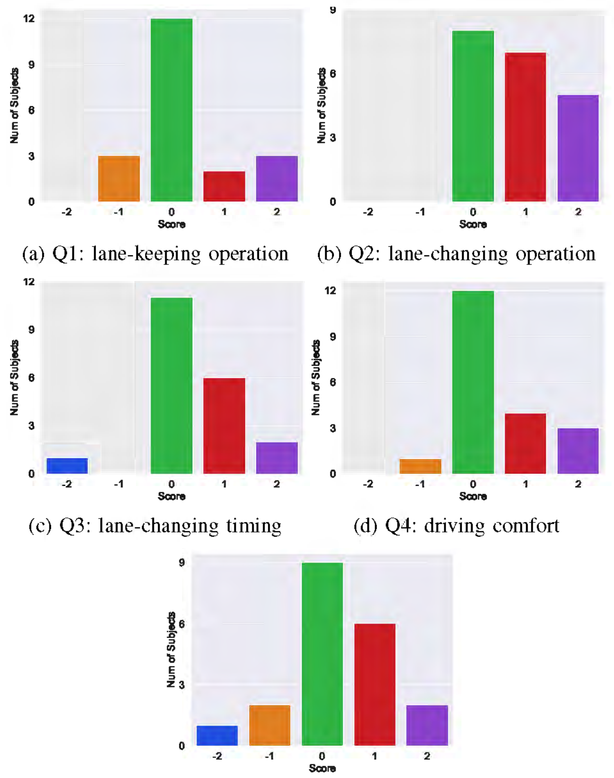
think it is reliable to ride in such an automated vehicle, while 15% have negative attitudes.

C. *Decision Efficiency*

In addition, we tested the average decision-making time of the learned policy in actual vehicle applications. Fig. 9 compares the single-step decision-making time and the receiving period of useful information. We can easily see that the average decision-making time is less than 4ms, with the maximum value less than 10ms. The receiving period of RTK information (via serial communication), steering wheel angle (via CAN), and surrounding vehicle information (via ethernet communication) are about 20ms, 8ms, and 50ms, respectively. The decision-making time is much shorter than the information receiving time interval and system response delay (about 100ms as shown in Fig. 4), indicating that the learned driving policy has great real-time decision-making capabilities.

V. Conclusion

In this paper, we designed a multi-lane driving task and the corresponding reward function to support RL-based policy learning. We used DSAC to learn an offline policy based on a simulation environment. Then, we applied the learned policy to a real car on a two-lane park road, and conducted objective and subjective experiments to verify the effectiveness of the learned policy. The results show that RL-based policy



(a) Q1: lane-keeping operation (b) Q2: lane-changing operation

(c) Q3: lane-changing timing (d) Q4: driving comfort

(e) Q5: dependability

Fig. 8. Subjective evaluation.

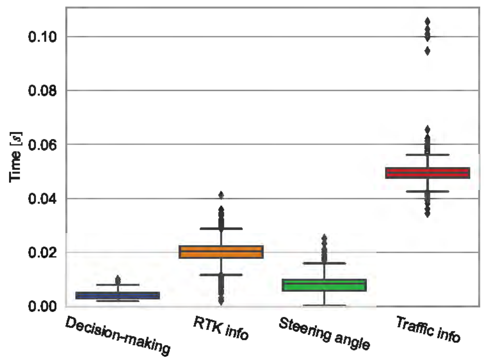


Fig. 9. Decision-making time and information receiving period. Each boxplot is drawn based on 100 values.

can smoothly complete maneuvers such as lane-keeping, lane-changing, and overtaking, so as to realize autonomous driving in response to different surrounding vehicles. Besides, the learned policy has high robustness to perceptual noise. The lateral deviation relative to lane center is less than 8cm, and the average single-step decision-making time is less than 4ms. Most subjects believe the learned driving policy can match or outperform human drivers in terms of lane-keeping operation, lane-changing operation, and driving comfort. Our work provides certain evidence for the feasibility of RL in real-world driving tasks.

References

[1] S. E. Li, *Reinforcement Learning for Decision-making and Control*. Springer, Berlin Heidelberg New York, 2022.

[2] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 509–522, 2011.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *the 4th International Conference on Learning Representations (ICLR)*. San Juan, Puerto Rico: ICLR, 2016.

[4] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 584–599.

[5] G. Li, S. Li, S. Li, and X. Qu, "Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient," *IET Intelligent Transport Systems*, 2021.

[6] E. Perot, M. Jaritz, M. Toromanoff, and R. De Charette, "End-to-end driving in a realistic racing game with deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Columbus, Ohio: IEEE, 2017, pp. 3–4.

[7] Q. Zou, H. Li, and R. Zhang, "Inverse reinforcement learning via neural network in driver behavior modeling," in *Intelligent Vehicles Symposium (IV)*. Changshu, Suzhou: IEEE, 2018, pp. 1245–1250.

[8] G. Li, S. Li, S. Li, Y. Qin, D. Cao, X. Qu, and B. Cheng, "Deep reinforcement learning enabled decision-making for autonomous driving at intersections," *Automotive Innovation*, vol. 3, no. 4, pp. 374–385, 2020.

[9] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *International Conference on Robotics and Automation (ICRA)*. Montreal, Canada: IEEE, 2019, pp. 8248–8254.

[10] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2034–2039.

[11] J. Duan, S. E. Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 297–305, 2020.

[12] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[13] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Transactions on Neural Networks and Learning Systems*, 2021, doi:10.1109/TNNLS.2021.3082568.

[14] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proceedings of the 34th International Conference on Machine Learning, (ICML 2017)*, Sydney, NSW, Australia, 2017, pp. 1352–1361.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 1861–1870.

[16] D. Krajzewicz, "Traffic simulation with sumo–simulation of urban mobility," in *Fundamentals of traffic simulation*. Springer, 2010, pp. 269–293.

[17] J. Duan, D. Yu, S. E. Li, W. Wang, Y. Ren, Z. Lin, and B. Cheng, "Fixed-dimensional and permutation invariant state representation of autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021, doi:10.1109/TITS.2021.3136588.