Proceedings of Machine Learning Research vol 242:1–13, 2024

Balanced Reward-inspired Reinforcement Learning for Autonomous Vehicle Racing

Zhen Tian 2620920Z@STUDENT.GLA.AC.UK
James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, U.K.

Dezong Zhao DEZONG.ZHAO@GLASGOW.AC.UK
James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, U.K.

Zhihao Lin 2800400L@STUDENT.GLA.AC.UK
James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, U.K.

David Flynn DAVID.FLYNN@GLASGOW.AC.UK
James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, U.K.

Wenjing Zhao WENJING.ZHAO@POLYU.EDU.HK
Department of Civil Environmental Engineering, Hong Kong Polytechnic University, Hong Kong, China

Daxin Tian DTIAN@BUAA.EDU.CN
School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

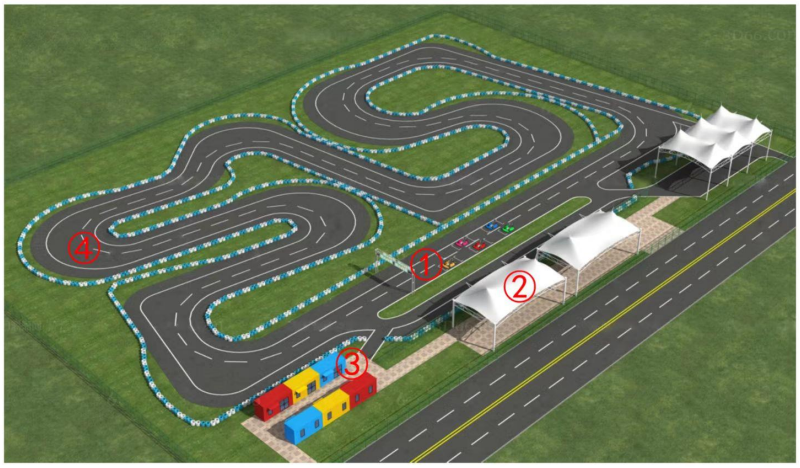
Abstract

Autonomous vehicle racing has attracted extensive interest due to its great potential in autonomous driving at the extreme limits. Model-based and learning-based methods are being widely used in autonomous racing. However, model-based methods cannot cope with the dynamic environments when only local perception is available. As a comparison, learning-based methods can handle complex environments under local perception. Recently, deep reinforcement learning (DRL) has gained popularity in autonomous racing. DRL outperforms conventional learning- based methods by handling complex situations and leveraging local information. DRL algorithms, such as the proximal policy algorithm, can achieve a good balance between the execution time and safety in autonomous vehicle competition. However, the training outcomes of conventional DRL methods exhibit inconsistent correctness in decision-making. The instability in decision-making introduces safety concerns in autonomous vehicle racing, such as collisions into track boundaries. The proposed algorithm is capable to avoid collisions and improve the training quality. Simulation results on a physical engine demonstrate that the proposed algorithm outperforms other DRL algorithms in in collision avoidance, achieving safer control during sharp bends, and higher training quality among multiple tracks.

Keywords: Autonomous vehicle racing, local planning, proximal policy optimization, balanced reward function.

1. Introduction

Autonomous vehicle racing has become an emerging field that combines the excitement of human vehicle racing and the state-of-the-art technologies in autonomous driving. Autonomous racing vehicles are expected to drive through complex tracks by exploring the limits of speed and decision-making. On the other hand, autonomous racing also escalates the entertaining value of the competition, by showcasing the amazing capabilities of autonomous vehicles. One of the main motivations for autonomous racing is to bridge the gap between the current level of autonomous driving and human driving at extreme limits.



① Start/End point ② Rest Area ③ Preparation Region ④ Racing Lane

Figure 1: Sketch of a closed-loop vehicle racing environment.

1.1. Motivation

Traditional vehicle racing is a challenging sport that requires reliable decision making, precise control, and robust perception. As illustrated in Figure 1, the racing track represents a complex and dynamic environment with varying properties such as track width, curvature, and surface conditions. In traditional vehicle racing, human drivers often encounter unexpected disturbances due to posture, viewing habits, and other factors [1]. Therefore, it is desirable to improve the driving safety by minimizing the affect of these disturbances during the racing. To this end, two main approaches are considered. The first approach involves optimizing the physical structures of the racing vehicle based on aerodynamics [2] [3]. The second approach involves designing robust control strategies from the online training. Recently, the growing interest of autonomous racing is demonstrated by several events such as Roborace [4, 5, 6], Formula Student Driverless [7], and Indy Autonomous Challenge [8, 9, 10]. In autonomous diving, the first stage is to obtain the driving environment information via perception. In autonomous racing competitions, the racing environment may equip with different perception levels. Therefore, to cope with diverse perception conditions, autonomous racing vehicles should not heavily depend on perception schemes.

Perception schemes for autonomous racing can be broadly classified into local perception and global perception. Global perception methods, such as [11], can assist the decision and planning of racing vehicles [11, 12]. Nonetheless, excessively depending on global perception introduces certain drawbacks. For example, sensor failure or communication loss may prevent the vehicles from obtaining global information. Therefore, it is required to develop methods that can enhance the racing vehicles to perceive local environments without global perception [**massa2020lidar**]. Model-based methods rely on predefined models or extra processes, such as Gaussian Process to quantify the uncertainty [13]. As a comparison, learning-based methods use data driven approaches to learn the optimal driving manner from data [14] [15]. Reinforcement learning (RL) is a powerful technique that learns optimal control commands from the training without global information [16, 17, 18]. Therefore, RL is capable to adapt to the local conditions of the environment. Recently, Deep reinforcement learning (DRL) has been developed for high dimensional complex tasks. To

enhance the autonomy level and reduce the reliance on perception, this paper would use DRL to achieve safe autonomous racing.

1.2. Related Works

State-of-the-art results of using DRL have been demonstrated in autonomous vehicles [19, 20, 21, 22]. Recently, a set of DRL algorithms with exceptional performance have attracted interest, such as Q-Learning [23], deep deterministic policy gradient (DDPG) [24], and proximal policy optimization (PPO) algorithms [25] [26].

In Q-Learning, the state-action value function is applied to determine the best action in a given state [27]. Correct actions are selected by Q-Learning in autonomous driving despite numerous safety constraints [28]. However, only simple tasks and scenarios were considered and tested in [28], without considering bends and more complex tasks. Moreover, another drawback of Q-Learning is the relatively low training efficiency [29].

DDPG uses deep neural networks to approximate the control policy [30]. With the suitability for handling high-dimensional data, multiple demonstrations of using DDPG have been given in autonomous driving [31]. For example, a DDPG model was proposed for safety driving within an end-to-end architecture [31]. Improved DDPG models have been proposed to improve training efficiency and results [32] [33]. However, the key issue is that DDPG gives an absolute result from the control policy. The absolute result hinders the exploration of more possible actions and restricts the adaptability in diverse driving scenarios.

PPO approximates the control policy in a probability distribution and offers fast strategy exploration improvements over DDPG [34]. PPO has been used to create control models for multi-agent driving scenarios [35]. PPO has been developed to generate smart driving strategies that balance safety and efficiency for crowded highway traffic [36]. In PPO, the traditional reward function uses a fixed ratio of averaged reward and current reward. The traditional reward function does not account for the influence of historical data on the current states in some cases. This leads to local high profits but global low profits. Hence, a reward function with a global perspective is needed.

1.3. Problem Statement and Contributions

DRL methods have greater potential than model-based methods in autonomous racing since they can better imitate the process of human decision making. However, The training results are unstable because the training guidance unbalances the local and global profits. To this end, a balanced reward-inspired PPO (BRPPO) is proposed in this paper. To optimize the future steps, a balanced reward function is proposed to consider both the historical and the prospective actions. The proposed reinforcement learning using PPO with the balanced reward function is illustrated in Figure 2. The whole algorithm is composed of offline training and real-time control. The decision network produces control commands in offline training, while the experience network generates real-time control commands. In offline training, the decision network learns to generate control commands based on the balanced rewards. The primary contributions are

- A balanced reward function is applied to deal with safety issues, such as collision during sharp bends. The collisions with sharp bends are avoided..

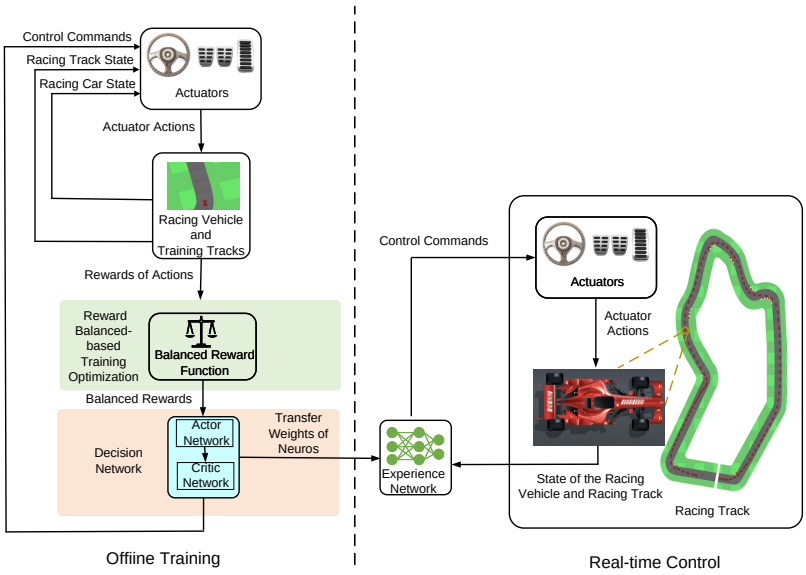


Figure 2: Diagram of the autonomous racing algorithm using the proposd BRPPO.

- The proposed BRPPO increases the training quality by balancing the historical and current rewards, enabling the autonomous racing vehicle to choose appropriate actions from the global perspective.

The rest of the paper is organized as follows: Section II introduces the decision network including the network structure and the control policy update process. Section III presents the details of balanced reward function. Section IV demonstrates the simulation results. Section V draws the conclusions.

2. Decision Network

The decision network is to generate safe and efficient control commands during training. The decision network consists of a set of actor-critic network that receives the balanced rewards of actions. The control policy in the actor-critic network compares the candidate control commands and choose the best one based on their relative advantages.

2.1. Network Structure

The decision network is composed of two actor-critic networks, which select actions based on the states of the racing vehicle. The actor-critic network is a neural network that integrates the control policy and the evaluation of control commands. The control policy selects control commands for less collisions. The evaluation of control commands is to estimate the relative advantage of the control commands. The relative advantage is served as a reference for updating the actor-critic network. The actor-critic network can be divided into the actor network (AN) and the critic network (CN). The AN and CN have similar structures, while the AN is to generate candidate control commands and the CN is to access the relative advantages for each candidate control command. The AN consists of an input layer, a series of convolutional layers, a linear layer and an output layer. The input

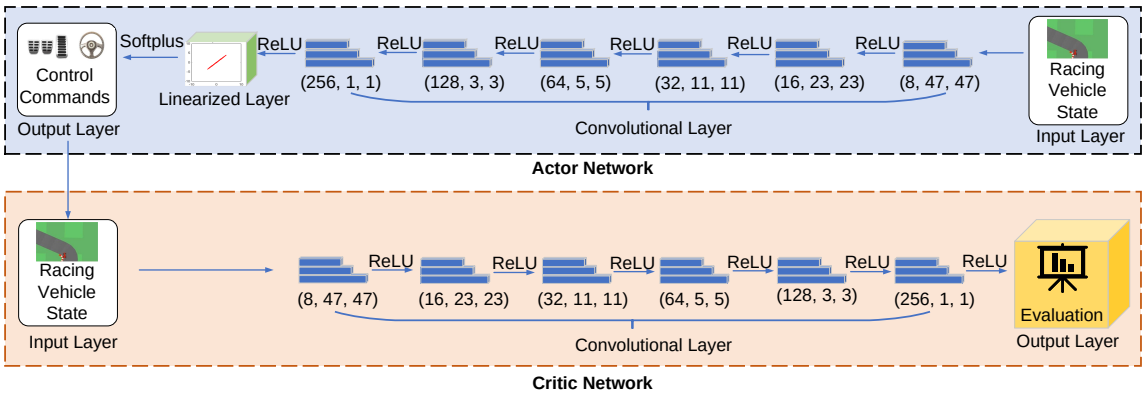


Figure 3: Structure of the actor-critic network.

layer receives the current state of the racing vehicle. The convolutional layers extract features from the driving states at each step. The linear layer adds linearity to the actor-critic network during the training. The output layer generates the control commands. The rectified linear unit (ReLU) layer applies the ReLU function to the output of the preceding layer. The ReLU function implements the operation $\max(x, 0)$ on each input tensor element, where x represents the input element. The objective of ReLU layer is to introduce non-linearity into the actor-critic network during training. The expression format of convolution layer CL is expressed as

$$CL = (A, B, C) \tag{1}$$

where A, B and C indicate the number of input channels, the number of output channels and the kernel size, respectively. The CN is composed of an input layer, a series of convolutional layers and an output layer. The input layer takes the output from AN and the current state of the vehicle as inputs. The convolutional layers extract features from the driving states and the corresponding control commands at each step. The output layer selects the best control commands based on their evaluation under the current state. The best control commands are then sent to the actuators. To generate a convincing evaluation, the relative advantage aims to reflect long-term advantages and spans a time period T. Hence, the CN compares the performance of a set of selected control commands with the average performance from the starting point t to $t+T$. The actor-critic network structure is illustrated in Figure 3.

2.2. Control Policy Update of the Decision Network

The control policy is determined by the weights of the neurons in the decision network. Therefore, the weights of the neurons should be adjusted to optimize the control policy. Figure 4 shows an example of a learning process involving a single racing sequence. The autonomous racing vehicle starts from the starting point with the maximum score. During the racing process, two types of losses including safety loss and efficiency loss are defined. When the autonomous racing vehicle reaches the finish point, a final score is calculated. Then, a score comparator compares the final score with a predefined expected score. If the final score is higher than the expected score, the

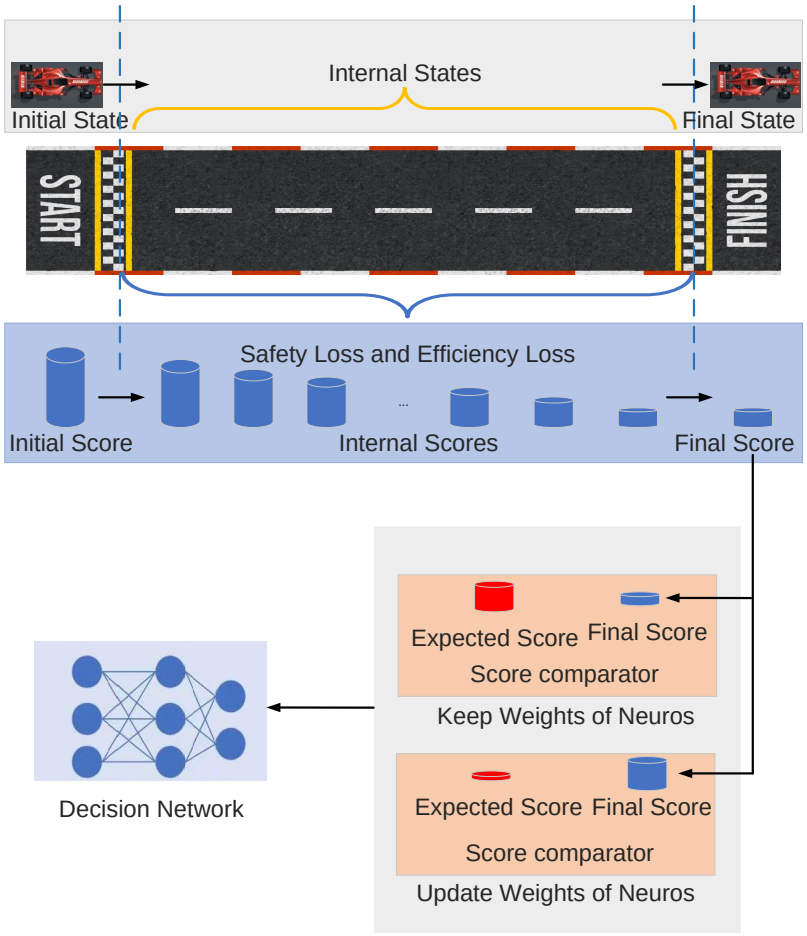


Figure 4: Control policy update of the decision network.

weights of the neurons in the decision network are updated. Otherwise, the weights of the neurons are not updated, as the performance does not meet the expected level.

If the final score is higher than the expected score, the weights of the neurons in the decision network are updated. Otherwise, the weights of the neurons are not updated, as the performance does not meet the expected level. When the racing car leaves the racetrack, it suffers significant safety losses, hindering the attainment of expected rewards. As decision sequences failing to reach the expected rewards are sieved out, the process of control policy updating prevents instances of the car veering off the track.

3. Balanced Reward Function

The reward function is used to evaluate the actions selected by the decision network. During the racing, the laptime and collision frequency are the two main factors that affect the performance of the competitors. The laptime reflects the effectiveness of the actions, while the collision frequency reflects the safety of the actions. Therefore, a good reward function for racing should guide the decision network to select actions that can minimize collisions with the track boundaries. However,

the traditional reward functions assigns equal weight to each step. The averaged reward is greatly influenced by previous high-reward actions. Therefore, the averaged reward is not able to balance the historical and current rewards. The averaged reward function is defined as

$$r_{\text{ave}} = 0.99 r_{\text{ave}} + 0.01 r_{\text{current}} \tag{2}$$

where r_{ave} and r_{current} are the averaged reward for historical states and the reward of the current state, respectively. Collisions during large and series bends at high speeds are the main safety concerns. The reward function should pay more attention to these critical single steps, which are called corner rewards. However, the averaged reward cannot focus on dangerous scenarios effectively, as the averaged reward gives equal weights to all historical steps. To address this issue, a hyper parameter is introduced to balance the average reward and the corner rewards. With the hyper parameter, a balanced reward function is proposed to consider both the historical and current rewards

$$r = (1 - \gamma) r_{\text{ave}} + \gamma r_{\text{current}} \tag{3}$$

where r represents the total reward under the current state of the racing vehicle, γ is a hyper parameter that directs the racing vehicle to prioritize random corners. With calibration, a value of γ that emphasizes most risky corners can be acquired. Therefore, γ promotes a safety-aware and forward-looking strategy, allowing the vehicle to predict possible dangers. Constraints on the learning speed are also required to restrained within a fair range during each update. To improve the stability in learning, a clipped surrogate objective is used to control the learning speed. The clipped surrogate objective prevents significant adjustments of neurons that might result in control policy divergence. The clipped surrogate objective is employed to update the policy network. The clipped surrogate objective is defined as

$$L_{\text{clip}} = \min(R * A, \text{clip}(R, 1 - \epsilon, 1 + \epsilon) * A) \tag{4}$$

where R represents the proportion of the new policy probability to the old policy probability, and the clip() function ensures that each component of the gradient is bounded between $1 - \epsilon$ and $1 + \epsilon$. A is obtained from the actor-critic networks, and ϵ is a self-defined hyper-parameter constraining the amplitude of alterations for the learning parameters during each iterative update.

4. Simulation Results

The simulations are designed to evaluate the safety of the BRPPO in different driving scenarios. To generalize the training results, racing tracks are randomly selected from the candidate tracks. The training quality of the BRPPO and two other DRL algorithms have been evaluated by their training scores on five racing tracks. The racing performance at critical bends, the number of collisions into track boundaries on five racing tracks with the BRPPO are compared and analyzed.

4.1. Simulation Environment

The training and testing environment employed is Box2D, which is a widely adopted open-source physics engine. The Box2D is designed to simulate and animate two-dimensional rigid-body dynamics [37]. In Box2D, the racing vehicle is modeled as a rigid body, consisting of several connected shapes, such as the chassis and wheels. The racing vehicle is connected to a controller that

generates control commands in terms of acceleration and steering based on a set of physical rules. Furthermore, the physical engine enables the simulation of suspension systems and enhances the fidelity of the simulation. In order to reduce the computing burden of BRPPO, a bicycle model [38] is used for the racing vehicle in Box2D

$$\dot{x} = V\cos(\varphi + \beta) \tag{5}$$

$$\dot{y} = V\sin(\varphi + \beta) \tag{6}$$

$$\dot{\varphi} = \frac{V}{l_r}\sin(\beta) \tag{7}$$

$$\dot{V} = a \tag{8}$$

$$\beta = \tan^{-1}(\frac{l_r}{l_f + l_r}\tan(\delta_f)) \tag{9}$$

where x and y represent the coordinates of vehicle's centre of mass, l_r is the length between the center of mass and vehicle's rear axle, l_f is the length between the center of mass and vehicle's front axle, β is the angle of the velocity with respect to the longitudinal axis of the vehicle, ψ represents the yaw angle. a and δ_f are chosen as the inputs. a is the vehicle longitudinal acceleration

$$a = F_{\text{throttle,max}} u_{\text{throttle}}/M \tag{10}$$

where $F_{\text{throttle,max}}$ and u_{throttle} are the maximum force of engine and the input level of throttle gate, respectively. M is the mass of the vehicle. δ_f the is steering angle given by

$$\delta_f = \delta_{\text{max}} u_{\text{steering}} \tag{11}$$

where δ_{max} is the maximum angle of steering and u_{steering} is the input of steering level. Therefore, the states of the vehicle can be changed by adjusting the inputs u_{steering} and u_{throttle} .

4.2. Evaluation of the Results

The BRPPO algorithm is compared against two popular benchmark algorithms, PPO and DDPG, to assess the racing safety and training quality. To ensure convincing simulation results, five random racing tracks with different curvatures are used in simulations. The five random racing tracks are correspond to Case 1 through Case 5, respectively.

As illustrated in Figure 5, both the PPO and DDPG were outperformed by the BRPPO in terms of scores and average score from Cases 1 to 5. The BRPPO algorithm achieves higher scores than the PPO and DDPG algorithms in each case. The main reason for the superior performance of the BRPPO algorithm is that the safety and efficiency losses are reduced during the racing. The safety loss is minimized by the balanced reward function, which enhances the training quality of the algorithm and prevents collisions. Figure 6 illustrates how BRPPO, PPO, and DDPG react to dangerous bends in a testing case. There are five bends from A to E in this case. Bend A has a high curvature, which makes it challenging to drive through. Bends B and C are the normal bends, which require moderate control. Bends D and E are close to each other, which increases the difficulty of steering. It can be seen that the BRPPO drives safer than the other two algorithms, as it travels within the boundaries and stays close to the inner side of curve when possible. At bend A, PPO deviates from the driving area, which causes a high safety loss. The DDPG follows the

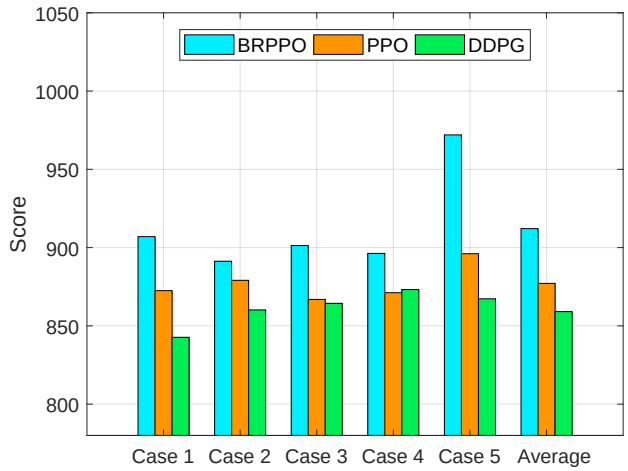


Figure 5: Scores for Cases 1 to 5 of using BRPPO, PPO and DDPG.

Table 1: Number of Collisions of Using BRPPO, PPO, and DDPG

	Number of Collisions		
Index of Tracks	BRPPO	PPO	DDPG
Track 1	**0**	2	4
Track 2	**0**	3	2
Track 3	**0**	4	4
Track 4	**0**	4	2
Track 5	**0**	4	4

outer side of the track, which increases its efficiency loss. At bend B, the DDPG also leaves the driving area, which leads to a high safety loss. At bend C and D, the BRPPO stays in the center of the track, which balances the safety and efficiency objectives. The DDPG moves closer to the inner side of the track boundary, which improves its efficiency performance. Bends C and D suggest that the BRPPO is willing to sacrifice some efficiency profits to avoid collisions. At bend E, both PPO and DDPG exit the driving area, which results in a high safety loss. Table I compares the times of accidents that BRPPO, PPO, and DDPG touch the boundaries from Cases 1 to 5. The results in Table I show that the BRPPO makes fewer collisions. Therefore, the proposed BRPPO algorithm is superior in handling the complexities in racing and achieves better overall performance.

5. Conclusion

This paper proposed a balanced reward-inspired PPO algorithm for autonomous racing, aiming to improve the training scores and driving performance of the racing vehicle. To enhance the attention to critic steps, a balanced reward function is used to balance the historical and current rewards during the training. The algorithm is trained and tested on a physical rules-based platform that

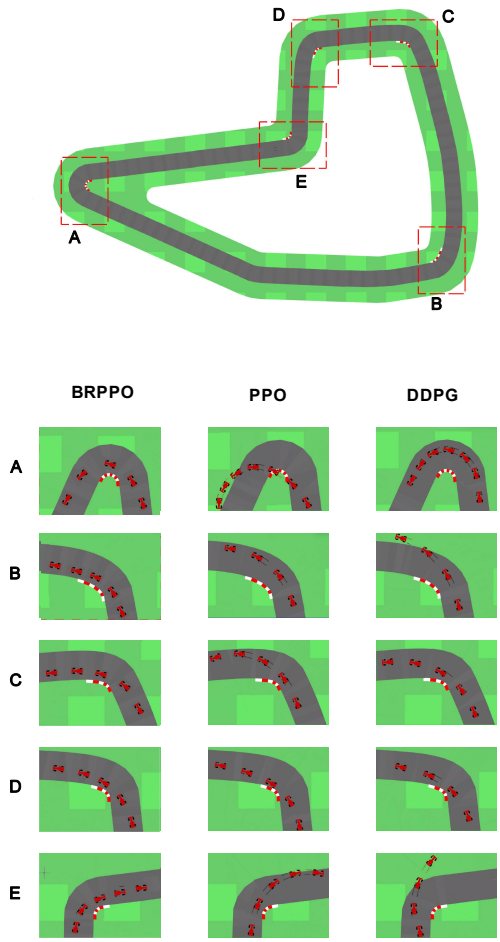


Figure 6: Driving performance of using BRPPO, PPO and DDPG in Case 5.

simulates the real driving environment. Comparisons among BRPPO and other two representative DRL algorithms were conducted, showing that the proposed algorithm outperforms in terms of higher average scores and fewer collisions. In the future, recognizing the limitations of the current scenario, which only considers a single racing vehicle and major racing factors, extensive research will be conducted in several aspects, including 1) extending the algorithm to team competitions of autonomous racing vehicles, 2) optimizing the racing process considering diverse objectives such as riding comfort, and 3) verifying the racing ability of BRPPO under more uncertain conditions.

References

[1] Wenjing Zhao et al. “Effects of collision warning characteristics on driving behaviors and safety in connected vehicle environments”. In: *Accident Analysis & Prevention* 186 (2023), p. 107053.

[2] Mehdi Imani Masouleh and David JN Limebeer. "Optimizing the aero-suspension interactions in a Formula one car". In: *IEEE Transactions on Control Systems Technology* 24.3 (2015), pp. 912–927.

[3] Pol Duhr et al. "Convex performance envelope for minimum lap time energy management of race cars". In: *IEEE Transactions on Vehicular Technology* 71.8 (2022), pp. 8280–8295.

[4] Johannes Betz et al. "A software architecture for an autonomous racecar". In: *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE. 2019, pp. 1–6.

[5] Johannes Betz et al. "Autonomous vehicles on the edge: A survey on autonomous vehicle racing". In: *IEEE Open Journal of Intelligent Transportation Systems* 3 (2022), pp. 458–488.

[6] Danilo Caporale et al. "Towards the design of robotic drivers for full-scale self-driving racing cars". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5643–5649.

[7] Juraj Kabzan et al. "AMZ driverless: The full autonomous racing system". In: *Journal of Field Robotics* 37.7 (2020), pp. 1267–1294.

[8] Chanyoung Jung, Andrea Finazzi, et al. "An Autonomous System for Head-to-Head Race: Design, Implementation and Analysis; Team KAIST at the Indy Autonomous Challenge". In: *arXiv preprint arXiv:2303.09463* (2023).

[9] Ayoub Raji et al. "er. autopilot 1.0: The full autonomous stack for oval racing at high speeds". In: *arXiv preprint arXiv:2310.18112* (2023).

[10] Johannes Betz et al. "Tum autonomous motorsport: An autonomous racing software for the indy autonomous challenge". In: *Journal of Field Robotics* 40.4 (2023), pp. 783–809.

[11] Florian Sauerbeck et al. "Learn to See Fast: Lessons Learned From Autonomous Racing on How to Develop Perception Systems". In: *IEEE Access* 11 (2023), pp. 44034–44050.

[12] Ayoub Raji et al. "Motion planning and control for multi vehicle autonomous racing at high speeds". In: *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*. 2022, pp. 2775–2782.

[13] Lukas Hewing, Alexander Liniger, and Melanie N Zeilinger. "Cautious NMPC with gaussian process dynamics for autonomous miniature race cars". In: *Proceedings of the European Control Conference*. 2018, pp. 1341–1348.

[14] Guofa Li et al. "Lane Change Strategies for Autonomous Vehicles: A Deep Reinforcement Learning Approach Based on Transformer". In: *IEEE Transactions on Intelligent Vehicles* 8.3 (2023), pp. 2197–2211. DOI: 10.1109/TIV.2022.3227921.

[15] Eshagh Kargar and Ville Kyrki. "Increasing the Efficiency of Policy Learning for Autonomous Vehicles by Multi-Task Representation Learning". In: *IEEE Transactions on Intelligent Vehicles* 7.3 (2022), pp. 701–710. DOI: 10.1109/TIV.2022.3149891.

[16] Jingwei Lu et al. "Event-Triggered Deep Reinforcement Learning Using Parallel Control: A Case Study in Autonomous Driving". In: *IEEE Transactions on Intelligent Vehicles* 8.4 (2023), pp. 2821–2831.

[17] Jingda Wu, Zhiyu Huang, and Chen Lv. "Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving". In: *IEEE Transactions on Intelligent Vehicles* 8.1 (2022), pp. 194–203.

[18] Zhaoxuan Zhu et al. "Safe model-based off-policy reinforcement learning for eco-driving in connected and automated hybrid electric vehicles". In: *IEEE Transactions on Intelligent Vehicles* 7.2 (2022), pp. 387–398.

[19] Le Lyu, Yang Shen, and Sicheng Zhang. "The Advance of reinforcement learning and deep reinforcement learning". In: *Proceedings of the IEEE International Conference on Electrical Engineering, Big Data and Algorithms*. 2022, pp. 644–648.

[20] Zhicong Liu et al. "A Methodology Based on Deep Reinforcement Learning to Autonomous Driving with Double Q-Learning". In: *Proceedings of the IEEE International Conference on Computer and Communications*. IEEE. 2021, pp. 1266–1271.

[21] Yasser H Khalil and Hussein T Mouftah. "Exploiting multi-modal fusion for urban autonomous driving using latent deep reinforcement learning". In: *IEEE Transactions on Vehicular Technology* 72.3 (2022), pp. 2921–2935.

[22] Lin Li, Wanzhong Zhao, and Chunyan Wang. "Pomdp motion planning algorithm based on multi-modal driving intention". In: *IEEE Transactions on Intelligent Vehicles* 8.2 (2022), pp. 1777–1786.

[23] Sina Alighanbari and Nasser L. Azad. "Deep Reinforcement Learning With NMPC Assistance Nash Switching for Urban Autonomous Driving". In: *IEEE Transactions on Intelligent Vehicles* 8.3 (2023), pp. 2604–2615. DOI: 10.1109/TIV.2022.3167616.

[24] Thomas Hickling, Nabil Aouf, and Phillippa Spencer. "Robust Adversarial Attacks Detection Based on Explainable Deep Reinforcement Learning for UAV Guidance and Planning". In: *IEEE Transactions on Intelligent Vehicles* 8.10 (2023), pp. 4381–4394. DOI: 10.1109/TIV.2023.3296227.

[25] Bile Peng et al. "Communication Scheduling by Deep Reinforcement Learning for Remote Traffic State Estimation With Bayesian Inference". In: *IEEE Transactions on Vehicular Technology* 71.4 (2022), pp. 4287–4300. DOI: 10.1109/TVT.2022.3145105.

[26] Kang Liu et al. "Reliable PPO-Based Concurrent Multipath Transfer for Time-Sensitive Applications". In: *IEEE Transactions on Vehicular Technology* 72.10 (2023), pp. 13575–13590. DOI: 10.1109/TVT.2023.3277712.

[27] Can Xu et al. "A Nash Q-learning based motion decision algorithm with considering interaction to traffic participants". In: *IEEE Transactions on Vehicular Technology* 69.11 (2020), pp. 12621–12634.

[28] Kyushik Min, Hayoung Kim, and Kunsoo Huh. "Deep Q learning based high level driving policy determination". In: *Proceedings of the IEEE Intelligent Vehicles Symposium*. 2018, pp. 226–231.

[29] Shixiang Gu et al. "Continuous deep Q-learning with model-based acceleration". In: *Proceedings of the International Conference on Machine Learning*. 2016, pp. 2829–2838.

[30] Kai Yang et al. "Towards robust decision-making for autonomous driving on highway". In: *IEEE Transactions on Vehicular Technology* 72.9 (2023), pp. 11251–11263.

[31] Giacomo Basile, Alberto Petrillo, and Stefania Santini. "DDPG based end-to-end driving enhanced with safe anomaly detection functionality for autonomous vehicles". In: *Proceedings of the IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering*. 2022, pp. 248–253.

[32] Yuansheng Dong and Xingjie Zou. "Mobile robot path planning based on improved DDPG reinforcement learning algorithm". In: *Proceedings of the IEEE International Conference on Software Engineering and Service Science*. 2020, pp. 52–56.

[33] Marawan Azmy Hebaish, Ahmed Hussein, and Amr El-Mougy. "Towards safe and efficient modular path planning using twin delayed DDPG". In: *Proceedings of the IEEE Vehicular Technology Conference*. 2022, pp. 1–7.

[34] Sanjna Siboo et al. "An Empirical Study of DDPG and PPO-Based Reinforcement Learning Algorithms for Autonomous Driving". In: *IEEE Access* 11 (2023), pp. 125094–125108. DOI: 10.1109/ACCESS.2023.3330665.

[35] Haoran Wei et al. "Mixed-autonomy traffic control with proximal policy optimization". In: *Proceedings of the IEEE Vehicular Networking Conference*. 2019, pp. 1–8.

[36] Fei Ye et al. "Automated lane change strategy using proximal policy optimization-based deep reinforcement learning". In: *Proceedings of the IEEE Intelligent Vehicles Symposium*. 2020, pp. 1746–1752.

[37] Ian Parberry. *Introduction to Game Physics with Box2D*. CRC Press, 2017.

[38] Michael Estrada, Sida Li, and Xiangyu Cai. "Feedback Linearization of Car Dynamics for Racing via Reinforcement Learning". In: *arXiv preprint arXiv:2110.10441* (2021).