This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

Efficient and Balanced Exploration-driven Decision Making for Autonomous Racing Using Local Information

Zhen Tian[1], Dezong Zhao[1], *Senior Member, IEEE*, Zhihao Lin[1], Wenjing Zhao[2], David Flynn[1], Yuande Jiang[3], Daxin Tian[4], *Senior Member, IEEE*, Yuanjian Zhang[5], *Senior Member, IEEE*, and Yao Sun[1], *Senior Member, IEEE*

***Abstract*—Autonomous racing has attracted extensive interest due to its great potential in self-driving at the extreme limits. Model-based and learning-based methods are widely used in autonomous racing. Out of which, model-based methods cannot cope with complex environments when only local perception is available. This limit can be overcome by the Proximal policy optimization (PPO), a typical learning-based method, which does not excessively rely on global perception. However, existing PPO faces challenges with low training efficiency in long sequences. To solve this issue, this paper develops an improved PPO by introducing a curiosity mechanism, a balanced reward function, and an image-efficient actor-critic network. The curiosity mechanism focuses on training on key segments, facilitating efficient short-term learning of the PPO. The balanced reward function adjusts rewards based on the complexity of racetracks, promoting efficient exploration of the control strategy during training. The image-efficient actor-critic network enhances the PPO to fast process the perceived information. Simulation results on a physical engine demonstrate that the proposed algorithm outperforms benchmark algorithms in achieving less number of collisions, higher peak reward with less training time, and shorter laptime among multiple testing racetracks.**

***Index Terms*—Autonomous racing, local information, proximal policy optimization, curiosity mechanism, balanced reward function.**

I. Introduction

CAR racing is a challenging and exciting sport that requires reliable decision making, precise control, and robust perception because of complex racetracks. As illustrated in Fig. 1, the racetracks are designed with a series of sharp bends, which makes safe driving more difficult at high velocity. To address the limitations of traditional car racing approaches, autonomous racing has been developed, which combines the excitement of human car racing and the state-of-the-art autonomous driving technologies. Compared to traditional car racing, autonomous racing can drive through complex tracks at the speed limits with high precision due to its superior decision-making capabilities. The capabilities of autonomous racing have been demonstrated in the Roborace [1]–[3], Indy Autonomous Challenge [4]–[6], and Formula Student Driverless [7].

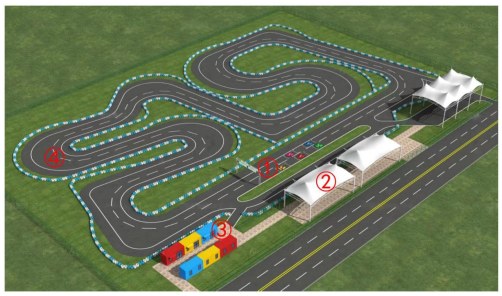


Fig. 1. Sketch of a closed-circuit car racing environment.

Global perception and local perception are both being applied to the autonomous racing. Out of which, local perception-based methods rely less on equipment and therefore are more cost effective. Perception-based decision making consists of model-based and learning-based methods. Learning-based methods are more promising because the global perception is not excessively used. For example, reinforcement learning (RL) is capable to adapt to the local conditions of the environment and generates optimal control commands [8]. Deep reinforcement learning (DRL), extending RL with deep neural networks (DNN) to handle complex functions, allows agents to learn from high-dimensional inputs like images. Existing DRL algorithms, such as the proximal policy optimization (PPO), perform well in short-term gaming scenarios. However, these algorithms still encounter challenges in the learning during the long-duration racing. To this end, a local perception-based, image-efficient, and balanced reward-orientated PPO with curiosity mechanism (PPO-C) is proposed in this paper, as illustrated in Fig. 2. The inputs of the decision network are the sequence of local images. An image-efficient decision network is proposed to process images and generate

*This work was sponsored in part by the EPSRC Innovation Fellowship under Grant EP/S001956/2, in part by the Royal Society-Newton Advanced Fellowship under Grant NAF\R1\201213, in part by the EPSRC Research Hub for Decarbonised Adaptable and Resilient Transport Infrastructures (DARe) under Grant EP/Y024257/1, and in part by the EPSRC Digital Twinning Research Hub for Decarbonising Transport.

[1]Z. Tian, D. Zhao, Z. Lin, D. Flynn and Y. Sun are with the School of Engineering, University of Glasgow, Glasgow, G12 8QQ, U.K. (e-mail: 2620920z@student.gla.ac.uk, dezong.zhao@glasgow.ac.uk, 2800400l@student.gla.ac.uk, david.flynn@glasgow.ac.uk, yao.sun@glasgow.ac.uk).

[2]W. Zhao is with the Department of Civil Environmental Engineering, Hong Kong Polytechnic University, Hong Kong, China (e-mail: wenjing.zhao@polyu.edu.hk).

[3]Y. Jiang is with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: jiangyd@chd.edu.cn).

[4]D. Tian is with the School of Transportation Science and Engineering, Beihang University, Beijing 100191, China (e-mail: dtian@buaa.edu.cn).

[5]Y. Zhang is with the College of Automotive Studies, Tongji University, Shanghai 201804, China (e-mail: 24093@tongji.edu.cn).

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

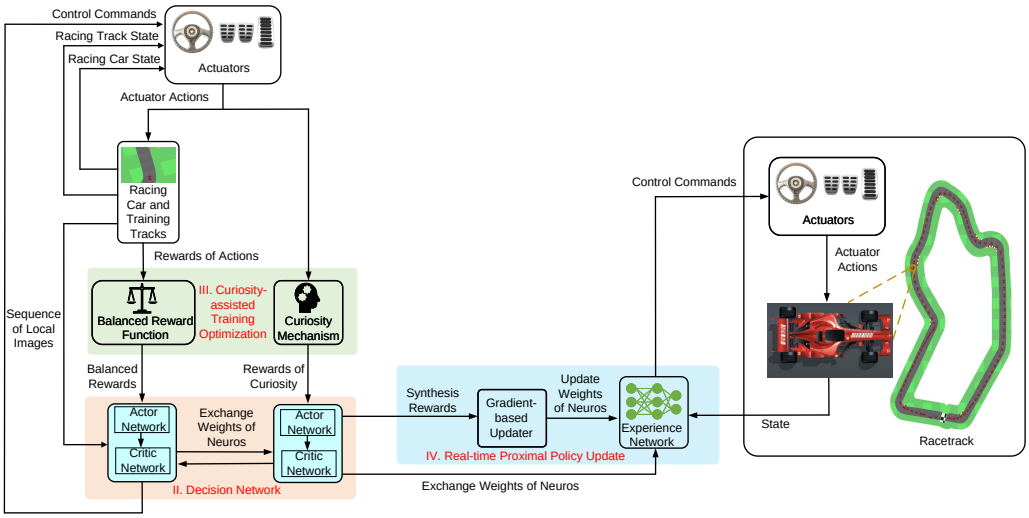


Fig. 2. Diagram of the autonomous racing algorithm using the curiosity-assisted proximal policy optimization.

safe control commands. The curiosity mechanism [9] uses intrinsic rewards to encourage the agents paying more attention to the steps with large prediction errors. Therefore, the agents can mitigate the uncertainties in local planning. Furthermore, a balanced reward function is proposed to consider both historical and prospective actions. The main contributions of this paper are as follows.

- Only the local perception is used to get images that combine the racing vehicle and the surrounding environment. Global perception is no more required in detecting the boundaries and the center line of the racing track.
- The time required to reach the saturation value of rewards is significantly reduced, and the collisions with sharp bends are avoided. The convergence of the training is improved over benchmark algorithms.
- Shorter laptime and less collisions are achieved by the proposed balanced reward function. The challenge of maintaining balanced exploration over long sequences is tackled by introducing the balanced reward function.

The rest of the paper is organized as follows. Section II summarizes the related works. Section III introduces the decision network. Section IV presents the details of curiosity-assisted training optimization. Section V elaborates the real-time proximal policy update mechanism. Section VI demonstrates the simulation results. Section VII presents the discussions. Section VIII draws the conclusions.

II. Related Works

A. Challenges in Autonomous Racing

In traditional car racing, human driving skills dominate the competition because unexpected disturbances are often encountered. To minimize the effects of these disturbances, two main approaches have been developed. The first approach aims to optimize the aerodynamics of the racing car, and the second approach is to design effective control strategies. Despite the demonstrated effectiveness, the first approach is restrained by the limited potential for improvement. For the majority of race cars, the aerodynamic models have been optimized to their maximum capacity. The drawback of the second approach lies in the absence of an experience-based decision-making mechanism. Therefore, the control performance cannot be effectively transferred to different tracks. Existing methods in autonomous racing are mainly ground in global perception and local perception. Global perception leverages comprehensive environmental data, the whole maps, and precise localization to provide a broad context for long-term planning [10], [11]. External sensors have been applied to global perception such as GPS, Inertial Measurement Unit (IMU), or Vehicle-to-Everything (V2X) communication. On the other hand, local perception focuses on real-time sensor data to detect and respond to immediate surroundings, ensuring dynamic object detection, short-term planning, and collision avoidance. Local perception uses onboard sensors, such as cameras and LIDARs [12]. For example, local perception is employed to perceive the surrounding road geometry and plan the vehicle speed in high-speed driving [13].

Global perception-based methods, predominately used in real world racing, heavily depend on specific perception conditions [6], [14], [15]. However, local perception-based methods are not bounded by specific perception conditions, reducing costs associated with global perception-based equipment. Therefore, local perception-based methods have gained popularity in autonomous racing [16], [17]. Model-based methods rely on pre-defined models or extra processes, such as Gaussian Process (GP) to quantify uncertainties [18]. However, model-based methods are incapable to cope with complex environments when only local perception is available. Model-based methods struggle in complex environments with only local perception due to their reliance on predefined planning and optimization rules. Without global information, these methods often lack the flexibility to handle unpredictable sections, as they may not obtain safe and efficient routes in unseen environments. A path-planning method is proposed in [19] that uses a path created by connecting the center lines on the straights and using clothoids between the center lines.

Table I
ENHANCED COMPARISON OF KEY FEATURES ACROSS DIFFERENT REINFORCEMENT LEARNING ALGORITHMS WITH REASONS FOR DIFFERENCES

Feature	PPO-C	DDPG	PPO	SAC	Reasons for Difference
Enhanced convergence consistency	✓		✓		PPO-C and PPO use a clipped objective function limiting excessive updates, enhancing training convergence consistency.
Ability to focus on complex segments	✓				Curiosity rewards enhance learning in complex scenarios against SAC and DDPG.
Adaptation to complex environments	✓				Curiosity rewards help PPO-C adjust strategy more effectively in complex conditions than other algorithms.
Improved data utilization efficiency	✓		✓		PPO-C and PPO update their learning multiple times per sample, improving efficiency.
Optimization of critical behaviors	✓		✓	✓	PPO-C targets critical areas through intrinsic curiosity, unlike standard PPO or SAC.
Promoting exploration in uncertainty	✓				Curiosity-based exploration targets high uncertainty areas ignored by other algorithms.
Advanced reward structure	✓				PPO-C uses prediction errors in rewards to accelerate learning, unlike other algorithms.

The forward center line is required for global perception. A minimum-time optimal control problem using the centerline of the racetrack is formulated in [20]. Furthermore, uncertainty quantification in Model-based methods, such as GP, may encounter challenges when the real racetrack differs significantly from the tracks used to define the uncertainties. As a comparison, learning-based methods learn the optimal driving manner from data [21]. DRL, an advanced learning-based method that leverages deep neural networks to approximate complex functions, enables agents to learn from locally perceptive images. Additionally, [22] secured the world championship in automobile racing by using the DRL. It demonstrated the outstanding capability of DRL to enhance both the safety and stability in autonomous racing. Furthermore, [23] proposes a DRL powered racing system that surpasses the quickest human driver among a dataset comprising more than 50,000 players.

B. Deep Reinforcement Learning

State-of-the-art results of using DRL have been demonstrated in autonomous cars [2]. Recently, a set of DRL algorithms with exceptional performance have attracted interest, such as deep deterministic policy gradient (DDPG), soft actor-critic (SAC) and PPO algorithms.

DDPG is an off-policy algorithm that uses deep neural networks to learn the control policy. With its suitability for handling high-dimensional data, multiple demonstrations of using DDPG have been presented in autonomous driving [24]. In particular, a DDPG model was proposed for safe driving within an end-to-end architecture [24]. Improved DDPG models have been proposed to enhance training efficiency [25]. The speed of racing cars could be accelerated by using DDPG, as demonstrated in [26]. A vision-based DDPG that considers driving safety at high speeds was proposed in [27]. In these studies, DDPG produces a definite control policy instead of a probability distribution of control policies. However, this definite control limits the exploration of other potential actions, implying that the decision may be satisfactory but not optimal. SAC is another off-policy model that incorporates a maximum entropy framework to enhance training robustness [28]. It has been shown to achieve higher average speeds than DDPG on multiple racetracks [29]. However, [29] focuses solely on optimizing average speed without considering other factors, such as reducing collisions with racetracks. Although SAC encourages exploration, it might not efficiently explore strategies to simultaneously minimize lap times and avoid collisions due to its undirected exploration. Furthermore, SAC's entropy term in the loss function sometimes leads to excessive exploration and slower convergence in complex scenarios.

The PPO depicts the control policy as a probability distribution, which facilitates faster exploration of strategies compared to DDPG [30]. Moreover, PPO uses a policy gradient-based method, achieving a stable equilibrium and providing assurance of its steadiness [31]. In contrast, off-policy algorithms are unstable and ineffective because they rely on training data that must be efficient under the current policy [32]. PPO has been used for generating driving strategies that balance safety and efficiency [33]. However, PPO aims to identify the most favorable steps for improvement while avoiding regression that could lead to performance degradation. In PPO, the agent may struggle to generalize its experiences across different states and actions, leading to slower convergence. Moreover, PPO is prone to falling into local optima, which increases the training time [34]. Furthermore, the training efficiency of PPO in complex environments is low [35].

In summary, current DRL algorithms encounter challenges in fully exploring the environment, unstable training in off-line algorithm, and exhibiting lower convergence speed. PPO addresses some of these challenges by employing probability distributions for exploration. However, the training efficiency of PPO diminishes in complex-environment tasks like racing due to the increased variability. Moreover, the inherent risk of collisions with track boundaries during perilous turns remains unresolved due to its averaged intention mechanism. Additionally, achieving balanced exploration is crucial in long sequences to effectively construct the probability distribution of PPO. To mitigate these issues, a balanced reward-orientated PPO with curiosity mechanism is proposed in this paper. The proposed curiosity mechanism directs the attention of PPO to

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

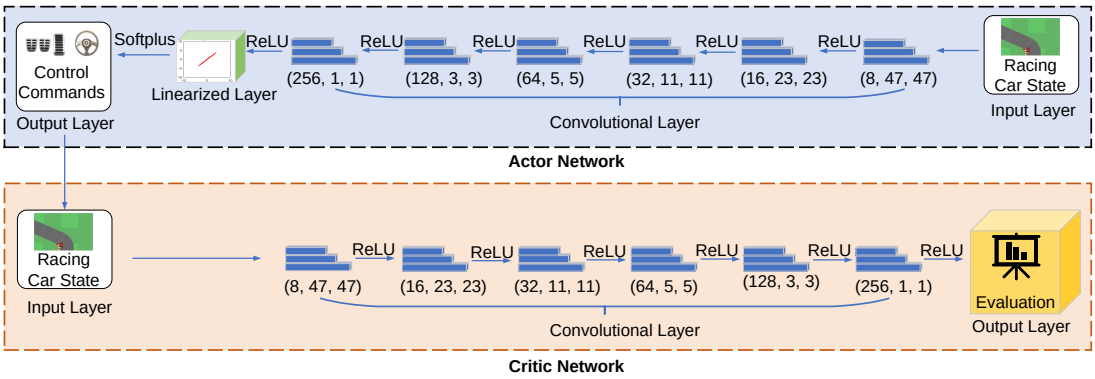


Fig. 3. Structure of the image-efficient actor-critic network.

critical short segments, thus enhancing the training efficiency. Furthermore, the balanced reward function facilitates balanced exploration from a global perspective. As a result, the low convergence speed and poor performance in crucial racing sections of PPO are addressed by introducing the curiosity mechanism and the balanced reward function.

The advantages of PPO-C compared to PPO, DDPG, and SAC are summarized in Table I. PPO-C uses intrinsic rewards to drive targeted exploration towards less-understood regions. The targeted exploration is particularly beneficial in complex environments such as racing, where standard rewards are sparse or less informative. Moreover, PPO-C excels in dynamically changing environments by continually adapting its policy to maximize both normal and intrinsic rewards. The adaptive learning fosters learning in crucial and difficult-to-navigate parts, optimizing critical behaviors, and promoting exploration based on state uncertainty.

Algorithm 1: Actor-Critic Network

Input: S_{t+1},r_t
Output: π_θ evaluated by the Critic-Actor Network
for *Each racing sequence* **do**
  for *m=1 to M* **do**
    for *t=m to T* **do**
      Run AN to receive an action $a_{(t)}$ using current policy
      Run CN to compute reward R_t,...,R_T
      Compute relative advantage A of $a_{(t)}$

$$A = R_m + \gamma R_{m+1} + ... + \gamma^T R_{m+T}$$

    end
  end
  Update π_θ according to A
end

III. Decision Network

The decision network is to generate safe and efficient control commands during training. The decision network consists of two sets of image-efficient actor-critic networks that receive the sequence of images, the balanced rewards of actions and the curiosity reward respectively. The control policy in the actor-critic network compares the candidate control commands and chooses the best one based on their relative advantages.

A. Network Structure

The aforementioned two actor-critic networks select actions based on the states of the racing car. Given the proven effectiveness of convolutional neural networks (CNN) in image classification [36], a series of convolutional layers are used to extract essential information from raw image data. The control policy selects commands to minimize collisions and laptime. Figure 3 illustrates the actor-critic network structure.

The actor-critic network comprises an actor network (AN) and a critic network (CN) in similar structures. The AN generates candidate control commands and the CN assesses their relative advantages. The AN consists of an input layer, convolutional layers, a linear layer, and an output layer. It processes the current state, extracts features, adds linearity for better representation learning, and generates control commands. The ReLU activation is used to introduce non-linearity. The CN is composed of an input layer, convolutional layers, and an output layer. It uses the AN output and current state as inputs, extracts features, and selects the best control commands based on their evaluation. The CN aims to reflect long-term advantages over a period T, comparing the performance of selected control commands with the average performance.

The 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.

To verify the image-efficient property of the proposed network, a comparison is made against the SqueezeNet [37]. The SqueezeNet is designed to achieve high accuracy with significantly fewer parameters and a smaller model size, making it theoretically suitable for processing imagingltraining data. However, when applied to the training in car racing, SqueezeNet achieves a peak reward of around 150, which is substantially lower than the peak reward of 900 obtained by the proposed network. This suggests that although SqueezeNet is efficient in parameters, it might not be as effective in

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

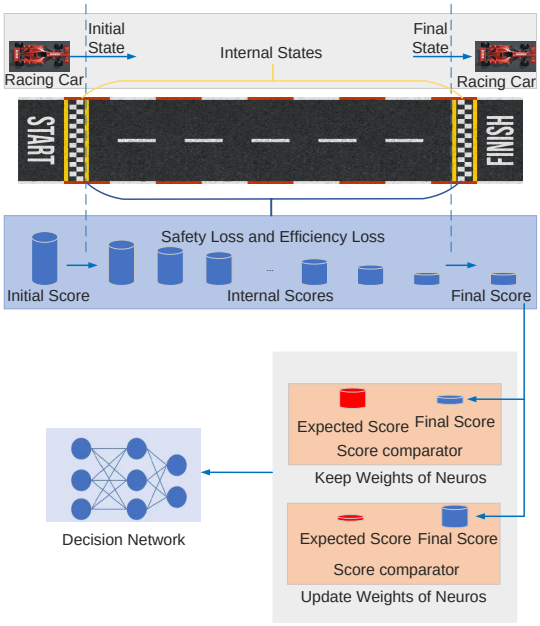


Fig. 4. Control policy update of the decision network.

handling the specific characteristics of racing images.The limited capacity and heavy reliance on 1×1 convolutions in SqueezeNet restrict its ability to capture intricate spatial relationships. Additionally, fine-grained details that are crucial for optimal performance in car racing may also be inadequately represented. The proposed network proves to be more successful in capturing spatial dependencies and making accurate decisions in the complex task of car racing.

B. 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 car starts from the starting point with the maximum score s_m. During the racing, two types of losses including safety loss L_s and efficiency loss L_e are defined. L_s increases as the distance to the track boundaries decreases. L_e is a constant value until the car completes the racing. When the autonomous racing car reaches the finish point, a final score is calculated.

The final score s_f is formulated as

$$s_f = s_m - L_s - L_e \tag{2}$$

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

IV. Curiosity-assisted Training Optimization

The curiosity-assisted optimization aims to enhance the training efficiency and the attention to dangerous sections, composing of the balanced reward function and the curiosity mechanism. The balanced reward function is to avoid collisions and reduce laptime during the training. The curiosity mechanism is to make optimal decisions in particular under hazard conditions.

A. Feature Encoding with CNNs

This paper uses local perception. Therefore, the input to the curiosity mechanism consists of a sequence of raw images, $\{I_t\}_{t=1}^{T}$, captured from the racing environment over a period of time, from t to T. The I_t represents the image at time step t. To extract meaningful features from these images, the CNNs are employed as the feature encoder. The CNNs learn to detect local patterns and features in the input images, reduce the spatial dimensions and provide translation invariance. Let θ_f denote the parameters of the feature encoder. At each time step t, the CNNs process the input image I_t and output a feature vector $F_{m,t}$:

$$F_{m,t} = \text{CNN}(I_t; \theta_f) \tag{3}$$

The encoded feature vector $F_{m,t}$ captures the relevant information from the input image I_t and serves as a compact representation of the racing environment at time step t.

B. Curiosity Mechanism

In RL, the agent is expected to pay attention to specific sections of racetracks. However, the traditional agent explores each part of the game with equal attention, indicating that no particular areas receive highlighted emphasis. Although a high averaged reward generally signifies good performance, safety issues may persist in dangerous corners due to unequal focus. Hence, establishing an attention-distribution mechanism is necessary. To diversify the focus across distinct sections, the curiosity space S_c is denoted by

$$S_c = \mid F_m - F_p \mid \tag{4}$$

where F_p denotes the predicted encoded features. S_c quantifies the discrepancy between the outputs F_m and F_p. A higher value of the discrepancy indicates a poorer understanding of the environment. Therefore, this value enables the agent to identify sections of the racetrack where its understanding is lacking and that require further exploration. By encouraging targeted exploration in these sections, the agent can efficiently gather data and refine its understanding of the environment. This targeted exploration also helps maintain a balance between exploration and exploitation. Therefore, the agent is ensured not to get stuck in suboptimal behaviors and continuously improves its performance.

At each time step t, assume that the action a_t, current state s_t and next state s_{t+1} are known. The output encoded features of the current state $F_{m,s}$ and the next state $F_{m,s+1}$ could be obtained via feature quantifier vectors

$$F_{m,s} = q(s_t, \theta_f) \tag{5}$$

$$F_{m,s+1} = q(s_{t+1}, \theta_f) \tag{6}$$

where $F_{m,s}$ denotes the current encoded features. $F_{m,s}$ is taken as the input to obtain the predicted encoded features of the next state $F_{p,s}$

$$F_{p,s} = \text{FM}(a_t, F_{m,s}) \tag{7}$$

where FM is the forward model to predict the feature representation of the next state. The curiosity reward space r_c could be obtained by

$$r_c = \beta \left\| F_{m,s+1} - F_{p,s} \right\|_2^2 \tag{8}$$

where β is a scaling factor obtained by calibration. The curiosity reward r_c plays a vital role in guiding the agent's exploration and enhancing its learning efficiency. While S_c quantifies the discrepancy between predicted and actual encoded features, r_c takes this discrepancy and directly incorporates it into the reward. The integration of curiosity into the reward function provides several advantages compared to using S_c alone. From a theoretical perspective, incorporating r_c directly into the reward modifies the RL objective to include an intrinsic motivation component. This modification can be formalized by augmenting the traditional reward function with r_c as an integrated reward. By directly influencing the agent's reward, r_c helps prioritize actions that reduce significant uncertainty, leading to more efficient learning. The agent receives immediate feedback by exploring uncertain states, which is reflected in the integrated reward. The integrated reward encourages a balanced approach to exploration and exploitation. This balance is crucial in RL, as it prevents the agent from focusing too much on curiosity (exploration) at the expense of task performance (exploitation).

By maintaining a constant exploration and learning, r_c helps the agent overcome unsatisfying sections associated with high discrepancy and facilitates continuous learning and improvement. Furthermore, β in the computation of r_c allows for the balancing of curiosity with the traditional reward. This ensures that the agent's exploration is flexibly guided by both curiosity and task-specific objectives.

Algorithm 2: Curiosity-assisted control policy update

1 Randomly initialize AN, CN, FM and Inverse Model (IM)
2 Initialize state s_0
3 Define the value of hyper-parameter α for the forward network (FN) and the backward network (BN)
4 **for** *m=1 to M* **do**
5 Use AN to obtain s_{m}, r_{m}, a_{m}, and $s_{\text{m+1}}$.
6 Compute r_{c} with (6).
7 Reconstruct r_{m} :
$$r_m = r_m + r_c.$$
8 Store s_m, r_m, a_m and s_{m+1} into the game sequence.
9 Compute A by Algorithm 1.
10 Compute the loss of FN L_F by
$$L_F = \left\| F_{\text{m,s+1}} - F_{\text{p,s}} \right\|_2^2.$$
11 Compute a_{p} by IM
$$a_p = \text{IM}(F_{\text{m,s}}, F_{\text{m,s+1}}).$$
12 Compute the loss of IM L_{B}
$$L_B = \left\| a_p - a_t \right\|_2^2.$$
13 Update BN and FN
$$\min_{\text{AN,FN,BN}} (1-\alpha) L_B + \alpha L_F.$$
14 Update π_θ in AN and CN using Algorithm 1.
15 **end**

C. Balanced Reward Function

The reward function is the feedback module that evaluates the actions generated by the decision network. During autonomous racing, the laptime and collision rates are the two major factors that evaluate the performance of the racing car. The laptime reflects the effectiveness of actions, while the collision frequency measures the safety of actions. Therefore, a good reward function for autonomous racing should guide the decision network to select actions that can avoid collisions with the track boundaries and reduce the laptime. However, the traditional reward functions assign equal attention to each step. The averaged reward is heavily 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

$$r_{\text{ave}} = 0.99 r_{\text{ave}} + 0.01 r_{\text{current}} \tag{9}$$

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 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. Moreover, the longer the sequence length, the less attention it pays to the performance of each single step. 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

$$r_b = (1 - \gamma * N_c) r_{\text{ave}} + \gamma * N_c * r_{\text{current}} \tag{10}$$

where γ represents a hyper parameter that directs the racing car to prioritize random corners. r_b is the balanced reward under the current state of the racing car. N_c is the number of corners. If there are lots of corners, the current decision is more crucial and thus the discount of historical reward becomes higher. r_{ave} is formulated as

$$r_{\text{ave}} = \sum_{i=1}^{N-1} s_f^i \tag{11}$$

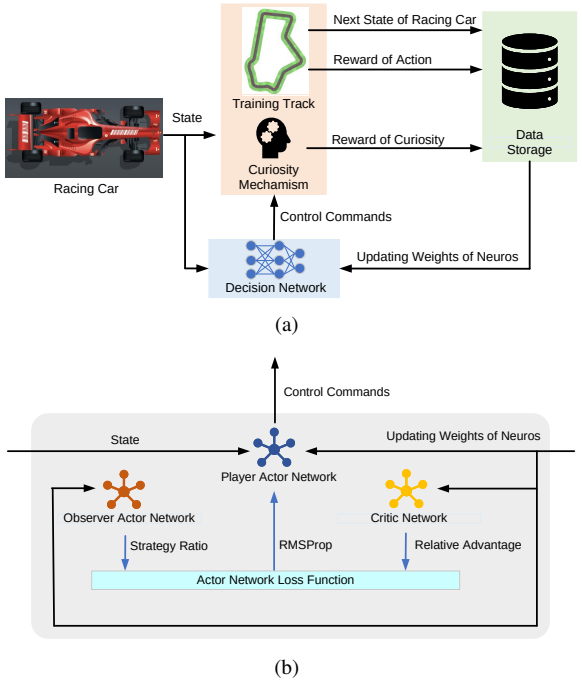


Fig. 5. Curiosity-assisted control policy update of the decision network. (a) General process of the control policy update. (b) Internal structure of the decision network.

where N is the number of current step. s_f^i is the final score at i^{th} step. r_{current} is equal to s_f at N step. Therefore, γ promotes a safety-aware and forward-looking strategy, allowing the car to pay attention to possible dangers.

Constraints on the learning speed are also required to be limited 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 lead to control policy divergence. The clipped surrogate objective is employed to update the policy network. The clipped surrogate objective is defined as

$$L_{\text{clip}} = \min(R * A, \text{clip}(R, 1-\epsilon, 1+\epsilon) * A) \qquad (12)$$

where R is the ratio of the new policy probability to the old policy probability, and the clip() function ensures that each component of the gradient is limited between $1-\epsilon$ and $1+\epsilon$. A is obtained from the decision networks, and ϵ is a self-defined hyper-parameter constraining the change for the weights of neuros during each iterative update.

D. Curiosity-assisted Control Policy Optimization

The update of control policy based on the curiosity mechanism is summarized in Algorithm 2. The IM is a component of the decision network that predicts the actions based on the current state and a target state. The FN is the forward network that implements the forward model using a neural network. BN is the backward network that learns the rationale of selecting actions from a target state and moving backward to the current state. Unlike traditional control policy updates, the curiosity-assisted control policy updates both the FN and the BN. The update of the FN and the BN is conducted by balancing the losses of the BN and the FN using a scaling factor α. The FN generates the curiosity reward, and the BN explores the sections that need high attention. Figure 5(a) illustrates the curiosity-based policy update. The generated data is stored in the data storage, which is the profit under the chosen actions based on the given state. The weights of the decision network are updated using both actions and curiosity rewards.

Figure 5(b) illustrates the internal structure of the decision network. The observer actor network compares the ratio between the updated strategy and the previous strategy to measure whether the update is proper. The critic network provides the relative advantages to access the values of control commands. $RMSProp$ is a process that helps train neural networks by adjusting the learning rate for each parameter.

The learning begins with an initial exploration. During the initial exploration, the agent randomly explores the racing environment. Therefore, the initial exploration allows the agent to form a preliminary understanding of the environment. After the initial exploration, the optimization is utilized to update the control policy network. With a continuous interaction with the environment, the agent has the potential to focus on critical areas. The curiosity mechanism enables the focused learning by directing the attention towards regions with higher potential for improvement. Throughout the learning, the decision results are evaluated against a set of predefined metrics, such as the laptime and collision occurrence frequency. By evaluating the driving performance during the training, the algorithm can be fine-tuned for various car racetracks.

E. Curiosity-based Training with Balanced Reward Function

To incorporate both the balanced reward r_b and the curiosity reward r_c, a dual decision network is employed. The network consists of two separate networks: the primary decision network D_p and the curiosity decision network D_c. Out of which, D_p is trained using the balanced reward r_b, which reflects the agent's performance in terms of safety and efficiency. D_c is trained using the curiosity reward r_c, which encourages exploration based on the discrepancy between predicted and actual encoded features. At each time step t, the input image I_t is processed by the CNN feature encoder to obtain the encoded features $F_{m,t}$. These encoded features are then given to both decision networks in generating their respective actions $a_{p,t}$ and $a_{c,t}$:

$$a_{p,t} = D_p(F_{m,t}; \theta_p) \qquad (13)$$

$$a_{c,t} = D_c(F_{m,t}; \theta_c) \qquad (14)$$

where θ_p and θ_c denote the parameters of D_p and D_c, respectively. The action a_t executed in the environment is determined by D_p. The training for the dual decision network is illustrated in Algorithm 3. During each training episode, the racing environment is reset, and the initial state s_0 is obtained. At each time step t, the input image I_t is processed by the CNN feature encoder to obtain $F_{m,t}$. $F_{m,t}$ are then given to D_p and D_c in generating their respective actions $a_{p,t}$ and $a_{c,t}$. The action a_t executed in the environment is determined by D_p. D_p is updated using the tuple $(s_t, a_{p,t}, r_{b,t}, s_{t+1})$, which

Algorithm 3: Curiosity-Driven Exploration with Balanced Reward Function

Randomly initialize CNN feature encoder with θ_f, D_p, D_c, FM, and IM.
Initialize state s_0.
Define the value of hyper-parameter α for balancing the losses.
for *epoch = 1 to N* **do**
  for *episode = 1 to M* **do**
    Reset racing environment and obtain initial state s_0.
    for *t = 0 to T-1* **do**
      Obtain input image I_t from state s_t.
      Compute encoded features $F_{m,t} = \text{CNN}(I_t; \theta_f)$.
      Generate actions $a_{p,t} = D_p(F_{m,t}; \theta_p)$ and $a_{c,t} = D_c(F_{m,t}; \theta_c)$.
      Execute action $a_t = a_{p,t}$ and observe next state s_{t+1}, reward $r_{e,t}$, and curiosity reward $r_{c,t}$.
      Obtain input image I_{t+1} from state s_{t+1}.
      Compute encoded features $F_{m,t+1} = \text{CNN}(I_{t+1}; \theta_f)$.
      Predict encoded features $F_{p,t} = \text{FM}(F_{m,t}, a_t; \theta_{fm})$.
      Compute curiosity reward $r_{c,t} = \beta \|F_{m,t+1} - F_{p,t}\|_2^2$.
      Reconstruct reward $r_t = r_{e,t} + r_{c,t}$.
      Store (s_t, a_t, r_t, s_{t+1}) into the replay buffer.
      Compute the loss of FM $L_F = \|F_{m,t+1} - F_{p,t}\|_2^2$.
      Predict action $a_{p,t} = \text{IM}(F_{m,t}, F_{m,t+1}; \theta_{im})$.
      Compute the loss of IM $L_I = \|a_{p,t} - a_t\|_2^2$.
      Update FM and IM by minimizing $(1-\alpha)L_I + \alpha L_F$.
      Update D_p and D_c using the stored experiences in the replay buffer.
    end
    Exchange weights between D_p and D_c.
    $\theta_p \leftarrow \theta_c$.
    $\theta_c \leftarrow \theta_p$.
  end
end

includes the balanced reward $r_{b,t}$. D_c is updated using the tuple $(s_t, a_{c,t}, r_{c,t}, s_{t+1})$, which includes the curiosity reward $r_{c,t}$. At the end of each training epoch, the weights of D_p and D_c are exchanged. This weight exchange allows the networks to share their learned knowledge and benefit from each other's experiences.

V. Real-time Proximal Control Policy Update

The real-time control policy update comprises the gradient-based control policy mechanism and the experience network. The experience network uses the gradient-based mechanism to adjust parameters and produce safe control commands.

A. Gradient-based Policy Update for Real-time Control

The PPO-C aims to learn the rules of choosing actions based on the states of the car and the local racing environment. Therefore, learning a precise policy in a short time is essential for effective RL. The gradient policy method is applied to learn the control policy more efficiently, enabling the experience network to update its driving strategy by leveraging the gradient of rewards.

A racing sequence includes a series of states. Denote N as the number of separated states. The control commands are generated by a probabilistic network for each state. The probability of choosing a proper action in a given state is written as $p_\theta(a_t|s_t)$. θ represents the parameters of the policy model. The training involves continuously updating the probabilities of different control commands. The racing sequential probability is formulated as

$$S = (p_\theta(a_1|s_1), p_\theta(a_2|s_2), ..., p_\theta(a_N|s_N)) \tag{15}$$

During the racing, the number of collisions with the track boundaries indicates the level of safety. The laptime indicates the level of racing capability. The probability of achieving a game sequence p_m in the m^{th} track is defined as

$$p_m = \sum_{t=1}^{N} p_\theta(a_t|s_t) \tag{16}$$

Assume the total number of racetracks is M, and R_m is the reward among the m^{th} track. Thus, to generate a probability distribution suitable for safe and efficient driving on different racetracks, a reward function is defined as

$$R_{\text{total}} = \sum_{m}^{M} R_m p_m \tag{17}$$

where R_{total} is the total reward among M tracks. As R_m is a fixed value for the sequence m, p_m should be adjusted to increase R_{total}. The radient descent method is an effective way to update the decision network towards the desired outcomes. The desired outcomes are defined as shorter laptime and collision avoidance. The gradient descent method is expressed as

$$\bigtriangledown f(x) = f(x) \bigtriangledown \ln f(x) \tag{18}$$

Lemma 1 is used to transform (14) to a more rigorous format.

Lemma 1. *For a differentiable function $f(x)$, the following equation holds:*

$$f(x)\frac{\mathrm{d}lnf(x)}{\mathrm{d}x} = f'(x)$$

Proof. Assume the initial conditions are

$$y = \ln f(x)$$
$$z = f(x) \tag{19}$$

Then we have

$$y = \ln z \tag{20}$$

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

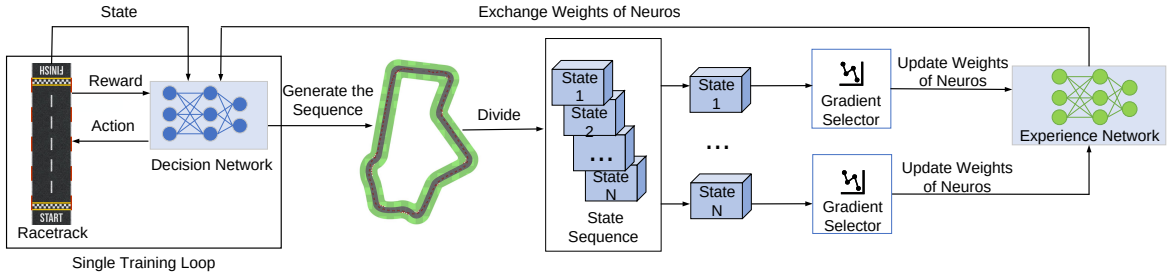


Fig. 6. The gradient-based control policy update process.

According to (17), the following equation is obtained

$$\frac{\mathrm{d}y}{\mathrm{d}z} = \frac{1}{z} \tag{21}$$

From (16) we have

$$\frac{\mathrm{d}z}{\mathrm{d}x} = f'(x) \tag{22}$$

It is seen that

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}z} \times \frac{\mathrm{d}z}{\mathrm{d}x} \tag{23}$$

According to (20), we have

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{1}{z} \times f'(x) \tag{24}$$

Afterwards, (21) could be further transferred to

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{1}{f_x} \times f'(x) \tag{25}$$

Combining $y = \ln f(x)$ with (22), the proof is finished with

$$f(x)\frac{\mathrm{d}\ln f(x)}{\mathrm{d}x} = f'(x) \tag{26}$$

■

With Lemma 1, (14) is further transferred to

$$\bigtriangledown R_{\text{total}} = \sum_{m}^{M} R_m p_m \bigtriangledown \ln p_m \tag{27}$$

Assuming that there is a long series of states, the probability of each racetrack p_m is extremely low and thus considered with the same small value. This small value is assumed to be in accordance with classical probability distribution, and (24) is further transferred to

$$\bigtriangledown R_{\text{total}} = \frac{1}{m}\sum_{m}^{M} R_m \bigtriangledown \ln p_m \tag{28}$$

Finally, each racing sequence could be expanded to N steps

$$\bigtriangledown R_{\text{total}} = \frac{1}{m}\sum_{m}^{M} R_m \sum_{t=1}^{N} \bigtriangledown \ln p_m(a_t|s_t) \tag{29}$$

The objective of (24) is to approach sequences associated with greater racing rewards. θ is updated by utilizing the rewards of each racing sequence. As illustrated in Fig. 6, the incorporation of a gradient-based mechanism significantly boosts the training efficiency by handling distinct states. Adhering to racing regulations, the racing reward is desired to reduce both laptime and achieve collision avoidance. The update process of θ among track m is formulated as

$$\theta = \theta + \alpha \bigtriangledown \log_e \pi_m(a_t|s_t) \tag{30}$$

where α is a parameter for the gradient exploration and π_m is the strategy network trained in the m^{th} track.

B. Control Policy Optimization of the Experience Network

The experience network has the same actor-critic network as the decision network. Therefore, the experience network keeps updating the actor-critic network parameters until the average reward meets the desired value. The five steps to implement the control policy update of the experience network are below.

- The historical game sequences are stored in the experience repla, which is a recorder of the rewards for different combinations of states and actions. The current racing environment, the car state and the reward are used to update the experience replay.
- Candidate commands are generated according to the current state, racing environment and control constraints.
- The relative advantages of candidate commands are estimated for every state-action pair in the generated data through the actor-critic network.
- The optimal control commands are generated according to the relative advantages.
- The average reward over the past training is assessed. If the average reward is higher than the desired value, the experience network is updated using a gradient-based policy. Otherwise, the car state is updated and returned to the first step for new iterations.

VI. Simulation Results

The simulations are designed to evaluate the safety and effectiveness of the PPO-C in different driving scenarios. To generalize the training results, racetracks are randomly selected from the candidate tracks. The training efficiency of PPO-C and three other DRL algorithms has been assessed in terms of training scores over various training epochs. The number of collisions with track boundaries and the lap time achieved by PPO-C have been compared and analyzed on 50 random racetracks. The racing performance at critical bends, trajectories, and variations of control levels are illustrated by four example cases.

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

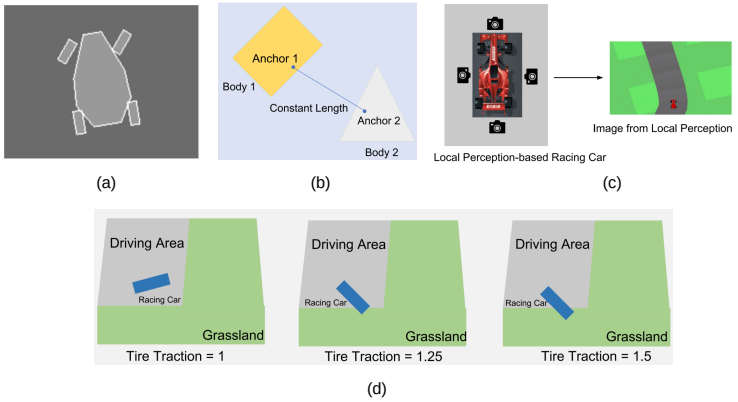


Fig. 7. The physical rule-based racing setup and test racetracks in the Box2D. (a) The physical model of the racing car in the Box2D. (b) The definition of fixed distance in the Box2D. (c) The local perception of racing car in the Box2D. (d) The various tire traction in the Box2D.

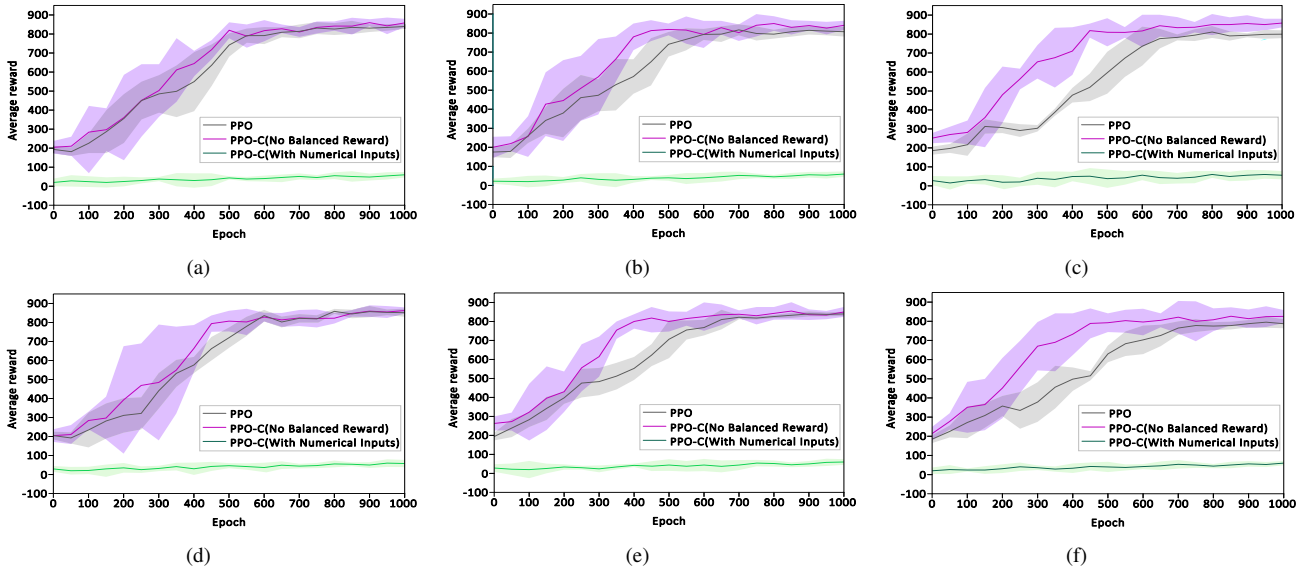


Fig. 8. The training curves of PPO-C without balanced reward, normal PPO, and PPO-C with numerical inputs across different minibatch sizes and scenarios. (a) The average reward curve with a minibatch size of 10 in scenario I. (b) The average reward curve with a minibatch size of 12 in scenario I. (c) The average reward curve with a minibatch size of 15 in scenario I. (d) The average reward curve with a minibatch size of 10 in scenario II. (e) The average reward curve with a minibatch size of 12 in scenario II. (f) The average reward curve with a minibatch size of 15 in scenario II.

A. Simulation Environmental Setup

The training and testing environment is Box2D, a widely used open-source physics engine designed to simulate and animate two-dimensional rigid-body dynamics [38]. In Box2D, the racing car is modeled as a rigid body with connected shapes, such as the chassis and wheels, resembling a real-world car. Figure 7(a) shows the car model in Box2D, which maintains a fixed distance between the body and tires, as exemplified in Fig. 7(b). Box2D also supports local perception, with cameras capturing images for the decision network, as illustrated in Fig. 7(c). Additionally, Box2D realistically models tire traction and body damping, considering car-track interactions, as illustrated in Fig. 7(d). Tire traction varies with the contact area, and damping influences stability, simulating real-world conditions. Moreover, Box2D uses collision filtering to manage collisions between the car and track boundaries, enabling realistic suspension system simulations and enhancing simulation fidelity.

B. Results and Analysis

1) Car Dynamics: In order to reduce the computing burden of PPO-C, a bicycle model is used for the racing car in Box2D [39]

$$\dot{x} = V\cos(\varphi + \beta) \tag{31}$$

$$\dot{y} = V\sin(\varphi + \beta) \tag{32}$$

$$\dot{\varphi} = \frac{V}{l_r}\sin(\beta) \tag{33}$$

$$\dot{V} = a \tag{34}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta_f)\right) \tag{35}$$

where x and y are the coordinates of car's centre of mass. l_r is the length between the center of mass and car's rear axle. l_f is the length between the center of mass and car's front axle. β is the angle of the velocity with respect to the longitudinal

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

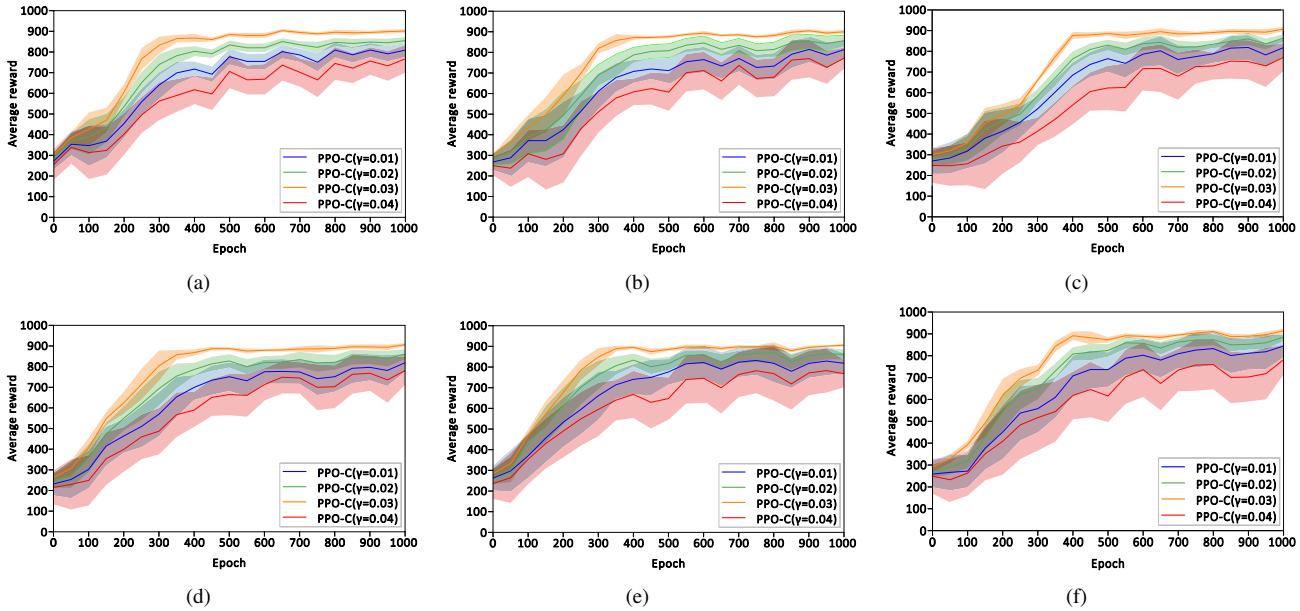


Fig. 9. The training curves of the PPO-C with γ from 0.01 to 0.04 across different minibathch sizes and scenarios. (a) The average reward curve with a minibatch size of 10 in scenario I. (b) The average reward curve with a minibatch size of 12 in scenario I. (c) The average reward curve with a minibatch size of 15 in scenario I. (d) The average reward curve with a minibatch size of 10 in scenario II. (e) The average reward curve with a minibatch size of 12 in scenario II. (f) The average reward curve with a minibatch size of 15 in scenario II.

axis of the car. ψ represents the yaw angle. a and δ_f are chosen as the inputs. a is the car longitudinal acceleration

$$a = F_{\text{throttle,max}} u_{\text{throttle}} / M \tag{36}$$

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 car. δ_f is the steering angle given by

$$\delta_f = \delta_{\max} u_{\text{steering}} \tag{37}$$

where $\delta_{\max}$ is the maximum angle of steering and u_{steering} is the input of steering level. Therefore, the states of the car can be changed by adjusting the inputs u_{steering} and u_{throttle}. The proposed algorithm is also applicable to the Ackerman model.

2) Scenario Description: During the training, the racing car starts at the initial point and the race is considered finished when it returns to the initial point. The car must avoid racetrack boundaries to ensure the safety, beginning with an initial speed of 0 and aiming to reach the final point as quickly as possible. This paper designs scenarios with varying degrees of racing aggressiveness to evaluate performance across different driving habits. The effectiveness of PPO-C is evaluated every 50 episodes. The car drives approximately 80 steps, typically encountering at least 6 curvy sections per racetrack.

- **Scenario I:** The car faces irregular racetracks with multiple curvy sections, increasing the difficulty of avoiding collisions. A penalty of 1 for efficiency at each step represents normal driving.
- **Scenario II:** The car has a higher penalty for efficiency of 1.5, demanding quicker completion during training, representing aggressive driving. All other settings are the same as in Scenario I.

Additionally, different minibatch sizes of 10, 12, and 15 are used to validate the effectiveness of PPO-C. Consistent performance across various minibatch sizes demonstrates the robustness of the algorithm, indicating its effectiveness is not batch-size dependent, making the results more reliable.

3) Fast Convergence by using the Image-based Curiosity Mechanism: Simulations are demonstrated in the training curves of image-based PPO-C without balanced reward, standard image-based PPO, and numerical features-based PPO-C across different minibatch sizes and scenarios. The numerical inputs used as embedded features include position, steering, and throttle openings. Fig. 8(a)-(f) plot the average reward against the epoch, showcasing learning performance over time.

In Scenario I, Fig. 8(a) shows that with a minibatch size of 10, image-based PPO-C significantly outperforms both PPO and numerical features-based PPO-C, achieving higher average rewards more rapidly and maintaining superior performance throughout training. Similarly, Fig. 8(b) and Fig. 8(c) depict minibatch sizes of 12 and 15, respectively, where image-based PPO-C achieves higher rewards earlier and consistently outperforms both PPO and numerical features-based PPO-C. In Scenario II, Fig. 8(d) with a minibatch size of 10 shows image-based PPO-C maintaining its superior performance, with higher average rewards across epochs. Figure 8(e) and Fig. 8(f) with minibatch sizes of 12 and 15, respectively, demonstrate that image-based PPO-C still outperforms both PPO and numerical features-based PPO-C. The poor performance of the numerical features-based PPO-C is due to the limited capability of CNNs to process numerical data effectively. Additionally, the numerical data can not reflect the distance of the racing car from the grasslands, contributing to the poor training results.

4) Reasoning Parameters of the Balanced Reward Function: Simulations are demonstrated in selecting the appropriate γ for a balanced reward function. To ensure sufficient

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

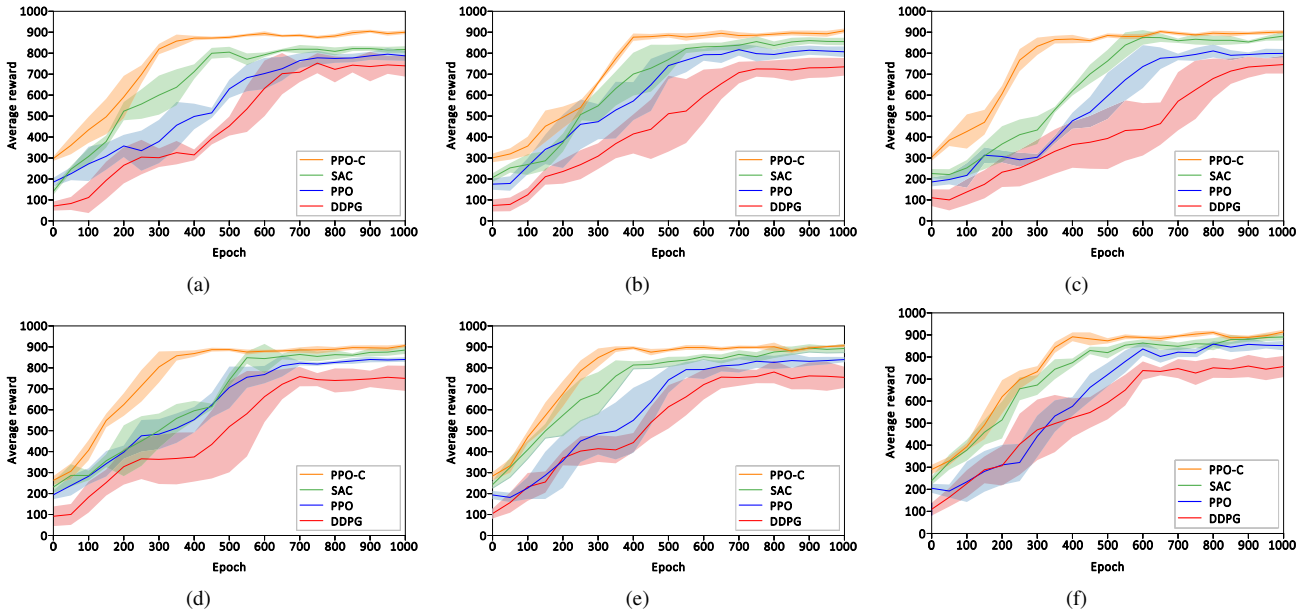


Fig. 10. The training curves of the PPO-C with other bench mark algorithms across different minibathch sizes and scenarios. (a) The average reward curve with a minibatch size of 10 in scenario I. (b) The average reward curve with a minibatch size of 12 in scenario I. (c) The average reward curve with a minibatch size of 15 in scenario I. (d) The average reward curve with a minibatch size of 10 in scenario II. (e) The average reward curve with a minibatch size of 12 in scenario II. (f) The average reward curve with a minibatch size of 15 in scenario II.

and convincing simulations, it is assumed that the historical reward still constitutes the major portion of the total reward. Therefore, in this paper, the minimum historical reward ratio is set at around 0.8. Considering that racetracks typically have approximately six corners in the Box2D environment, we select the maximum $\gamma = 0.04$:

$$\gamma = \frac{(1 - \text{historical reward})}{\text{number of corners}} = \frac{(1 - 0.8)}{6} = 0.036 \approx 0.04$$

The other three candidate values for γ = 0.01, 0.02, and 0.03, respectively. To verify the generalization of the most suitable parameter for learning, three different minibatch sizes are used: 10, 12, and 15. Additionally, to confirm the adaptability of the best parameter across varied driving styles in racing, two different scenarios are employed to determine the most appropriate parameter.

Figure 9 displays the training curves of PPO-C with various values of γ (ranging from 0.01 to 0.04) across different minibatch sizes and scenarios. Figure 9(a) to Fig. 9(f) represent the following conditions: Fig. 9(a) to Fig. 9(c) are simulation results with minibatch sizes of 10, 12, and 15 in Scenario I, respectively; Fig. 9(d) to Fig. 9(f) are simulation results with minibatch sizes of 10, 12, and 15 in Scenario II, respectively. Across all the test cases, there is a consistent trend of increasing average rewards with the number of epochs, generally stabilizing between 600 and 1000 epochs. Notably, the PPO-C with $\gamma = 0.03$ tends to perform better across multiple settings. Curves with $\gamma = 0.03$ consistently achieve higher average scores and show more stability as training progresses. For instance, in Fig. 9(a) and Fig. 9(d) with a minibatch size of 10, curves with $\gamma = 0.03$ demonstrate superior performance compared to other values. Similarly, in Fig. 9(b), Fig. 9(c), Fig. 9(e), and Fig. 9(f) with larger minibatch sizes, the curves with $\gamma = 0.03$ continue to outperform the others, achieving higher scores and smoother trends. The variability of the reward curves decreases with larger minibatch sizes, showing smoother trends for minibatch sizes of 15 compared to those of 10. Overall, $\gamma = 0.03$ is identified as the best-performing configuration across the various scenarios and minibatch sizes. Through comparisons with other benchmark algorithms, $\gamma = 0.03$ will be applied.

5) Comparison of Training Curves among Different Benchmark Algorithms: Figure 10 displays the training curves of PPO-C compared with other benchmark algorithms across different minibatch sizes and scenarios. In Fig. 10(a) with a minibatch size of 10 in Scenario I, the PPO-C outperforms other algorithms consistently, achieving higher average scores and demonstrating more stability, especially noticeable after 400 epochs. In Fig. 10(b) with a minibatch size of 12 in Scenario I, the PPO-C shows superior performance, rising more sharply and stabilizing at a higher average score. Figure 10(c) with a minibatch size of 15 in Scenario I shows the PPO-C continuing to outperform other algorithms, achieving higher average scores more quickly and maintaining steady improvement. In Scenario II, Fig. 10(d) with a minibatch size of 10, PPO-C remains the top performer, with its curve rising rapidly and stabilizing at a higher level. Figure 10(e) with a minibatch size of 12 in Scenario II shows PPO-C outperforming SAC, PPO, and DDPG, achieving higher scores and showing less variability. Finally, in Fig. 10(f) with a minibatch size of 15 in Scenario II, the PPO-C maintains its lead, achieving higher average scores and exhibiting smoother trends. Overall, the PPO-C consistently demonstrates superior performance across various scenarios and minibatch sizes, achieving higher average scores and showing more stability compared to SAC, PPO, and DDPG, underscoring its robustness and adaptability in different training conditions.

Table II
AVERAGE LAPTIME AMONG 50 RACETRACKS

		Laptime			
		PPO-C	**SAC**	**PPO**	**DDPG**
Scenario I	Minibatch 10	**24.13**	26.23	26.73	25.32
	Minibatch 12	**23.52**	25.14	26.32	25.76
	Minibatch 15	**22.91**	24.36	25.96	24.74
Scenario II	Minibatch 10	**23.93**	24.12	25.33	24.76
	Minibatch 12	**22.26**	23.88	24.74	24.56
	Minibatch 15	**22.44**	22.56	23.77	24.25

Table III
AVERAGE NUMBER OF COLLISIONS AMONG 50 RACETRACKS

		Number of Collisions			
		PPO-C	**SAC**	**PPO**	**DDPG**
Scenario I	Minibatch 10	**0.54**	1.26	2.16	2.86
	Minibatch 12	**0.46**	0.82	1.72	2.76
	Minibatch 15	**0.42**	0.56	1.66	2.74
Scenario II	Minibatch 10	**0.64**	0.72	2.68	3.24
	Minibatch 12	**0.52**	0.62	2.46	3.16
	Minibatch 15	**0.48**	0.66	2.06	3.22

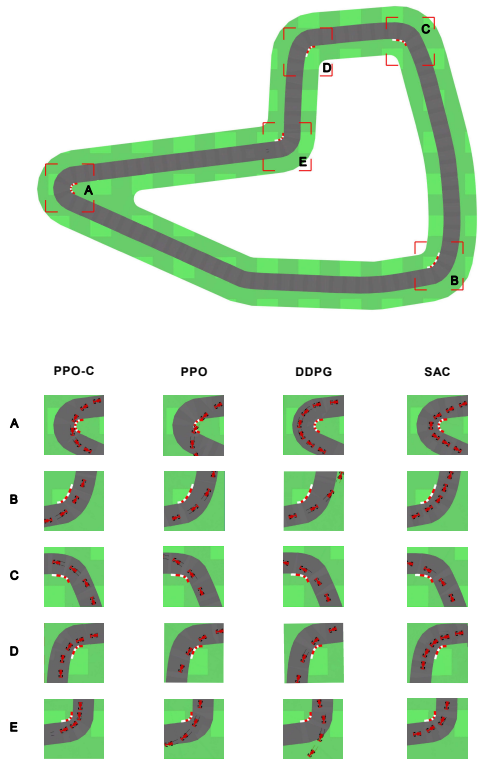


Fig. 11. Driving performance of using PPO-C, PPO, DDPG and SAC in an Example Case.

6) Evaluation of the Results: The PPO-C algorithm is compared against three benchmark algorithms recently used in racing, PPO, DDPG and SAC. Table II compares the laptime of the PPO-C and other benchmark algorithms among 50 random racetracks across different racing conditions and minibatch sizes. In normal racing, PPO-C records the minimum laptime of 24.13, 23.52, and 22.91 for Minibatch 10, 12, and 15 respectively. In Aggressive Racing, PPO-C continues to lead with minimum number of collisions of 23.93, 22.26, and 22.44 for the same minibatch sizes. SAC remains competitive, typically ranking second, while PPO and DDPG exhibit longer laptime. These results highlight PPO-C's superior capability in minimizing the laptime, demonstrating its effectiveness in both normal and aggressive racing scenarios.

Table III compares the number of collisions of the PPO-C and other benchmark algorithms among 50 racetracks across different racing conditions and minibatch sizes. PPO-C achieves a minimum number of collisions of 0.54, 0.46, and 0.42 for Minibatch 10, 12, and 15 respectively in normal racing, and 0.64, 0.52, and 0.48 in aggressive racing. SAC consistently ranks second in performance, followed by PPO and DDPG with higher collision rates. These results suggest that PPO-C excels in minimizing collisions across varying racing dynamics and minibatch sizes.

Figure 11 illustrates how PPO-C and the other benchmark algorithms react to dangerous bends in an example case. There are five bends from A to E in this case. Bend A has a high curvature, making it challenging to drive through. Bends B and C are normal bends, requiring moderate control. Bends D and E are close to each other, increasing the difficulty of steering. It can be seen that PPO-C demonstrates safer and smarter driving than the other three algorithms, as it travels within the boundaries and stays close to the inner side of the curve when possible. In bend A, PPO deviates from the driving area, causing a high safety loss. DDPG follows the outer and middle side of the track, increasing its efficiency loss. SAC drives along the inner track, decreasing the time consumption. In bend B, DDPG also leaves the driving area, leading to a high safety loss. In bends C and D, PPO-C stays in the center of the track and drives along the track boundary, respectively, balancing the safety and efficiency objectives. DDPG and SAC move closer to the inner side of the track boundary, improving their efficiency performance. Bends C and D suggest that PPO-C is willing to sacrifice some efficiency profits to avoid collisions. In bend E, both PPO and DDPG exit the driving area, resulting in a high safety loss.

In Table IV, the comparison of average speed and average lateral acceleration across five corners for different DRL algorithms is illustrated. For average speed, PPO-C demonstrates higher levels in four out of five corners compared to SAC, PPO, and DDPG. This suggests that PPO-C adjusts its speed effectively on straight sections before entering corners, indicating a balanced approach that takes into account the connection between straight sections and corners. Higher speeds in straight sections can contribute to maintaining competitive performance while ensuring stability during cornering, as evidenced by PPO-C's consistent higher speeds.

Regarding average lateral acceleration, PPO-C generally exhibits lower acceleration levels in corners A to D compared to other algorithms. Lower lateral acceleration indicates smoother

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

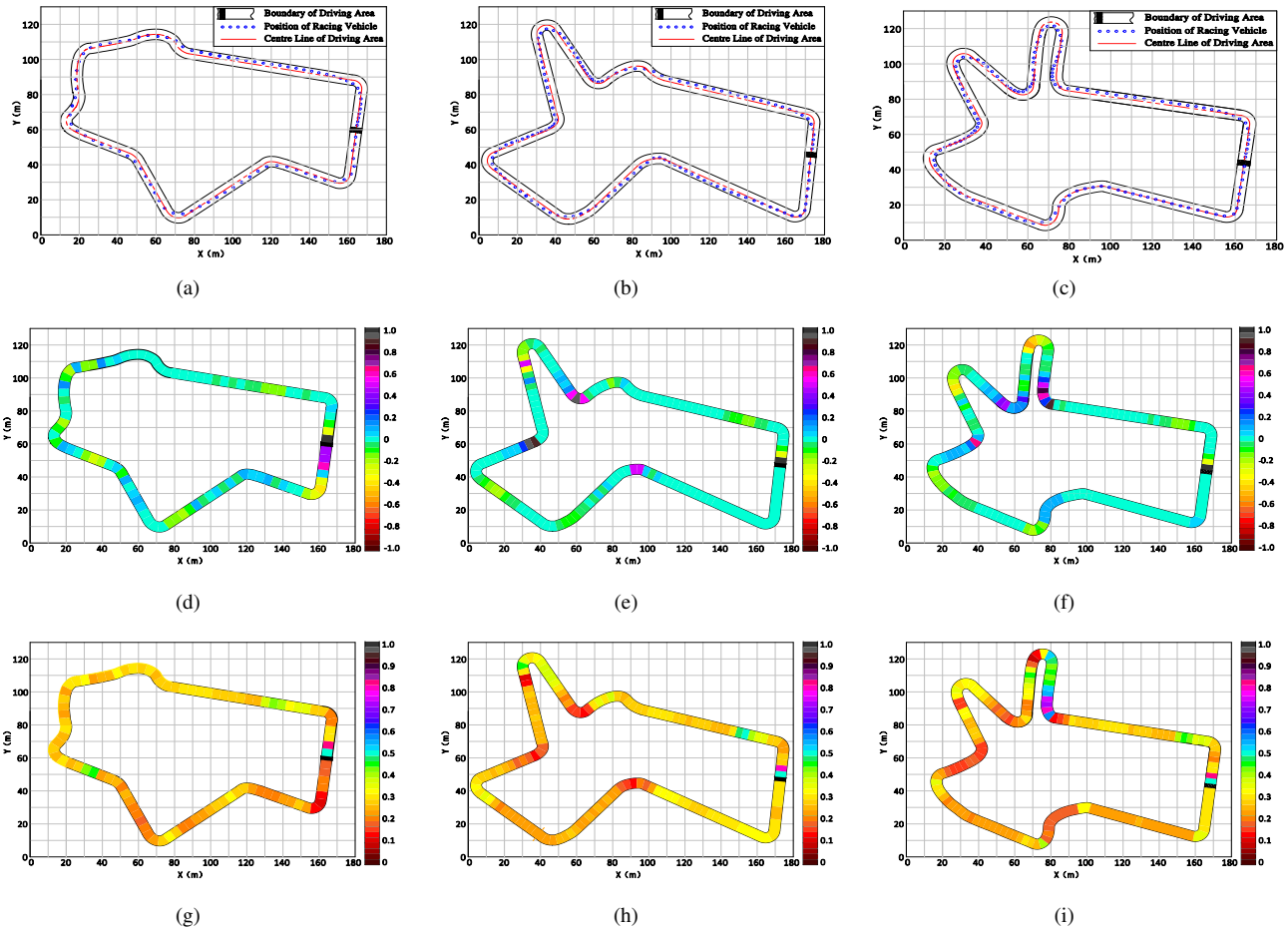


Fig. 12. The driving performance and control levels of three sample autonomous racing cases. (a)-(c) are the trajectories of cases 1-3, respectively; (d)-(f) are the steering angles of case 1-3, respectively; (g)-(i) are the throttle openings of cases 1-3, respectively.

and more stable driving, reflecting the ability of PPO-C to make balanced decisions and maintain stability throughout the track. Notably, corners D and E, being closely positioned, highlight a strategy where acceleration is applied in the first corner and deceleration in the subsequent one, optimizing control and speed management through successive turns.

Conversely, DDPG shows lateral deceleration across most corners, implying potentially higher speeds on straight sections followed by necessary deceleration in corners to maintain control. However, the high lateral acceleration in corner E for DDPG suggests challenges in maintaining control within the track boundaries, leading to instances where the vehicle exceeds the driving area.

7) Driving Performance and Control Levels in Three Sample Cases: There are no shortcuts in the testing tracks, ensuring the algorithm cannot exploit any contingencies. The testing tracks feature sharp or multiple curves, increasing difficulty. The racing car starts from the center of the starting point and aims to reach the end point quickly. Figure 12 demonstrates that the racing car follows a safe and efficient trajectory within the feasible racetracks. Fig. 12(a) to Fig. 12(c) show the trajectories of Case 1 through Case 3, respectively, with a color bar indicating steering and throttle opening ranging from -1 to 1. Fig. 12(d) to Fig. 12(f) illustrate the steering angles of Case 1 through Case 3, respectively, and Fig. 12(g) to Fig. 12(i) show the throttle openings for these cases.

In Fig. 12(a), the car deviates from the inner track boundary to avoid collisions. In Fig. 12(b) and Fig. 12(c), the car prefers the inner side of most curves to minimize lap time. These results show that PPO-C effectively balances safety and efficiency. Fig. 12(d) indicates that the car maintains its steering within -0.2 to 0.2 on curvy roads without large bends. In Fig. 12(e), the car exhibits both high steering around large bends and minor adjustments around consecutive bends. In Fig. 12(f), the car adjusts its steering angle more frequently due to larger and more consecutive bends, preferring slight steering on small bends and sharper steering on large bends. Fig. 12(g) illustrates that the car briefly increases its throttle opening when leaving curvy sections. In Fig. 12(h), the car reduces its throttle opening when passing the second bend in a series. In Fig. 12(i), the car changes its throttle more frequently due to larger and more consecutive bends, maintaining a throttle opening around 0.3 on straight roads. Thus, the throttle control strategy involves steady acceleration on small bends and more pronounced adjustments for a series of bends.

To illustrate the advantages of the proposed algorithm, this paper benchmarked against recent studies in Table IV. The DRL in [40] demonstrates enhanced training efficiency but

Table IV
COMPARISON OF AVERAGE SPEED AND AVERAGE LATERAL ACCELERATION IN 5 CORNERS

	Metrics			
	PPO-C	**SAC**	**PPO**	**DDPG**
Average Speed (m/s)				
Corner A	**20.69**	19.23	17.23	18.86
Corner B	**24.45**	23.67	19.36	22.72
Corner C	30.67	32.98	26.46	28.23
Corner D	**20.87**	20.45	18.34	19.66
Corner E	**15.99**	15.56	14.65	9.43
Average Lateral Acceleration (m/s^2)				
Corner A	**3.95**	4.37	5.76	-2.23
Corner B	**6.32**	7.26	6.30	-4.22
Corner C	**8.57**	9.37	9.15	-0.28
Corner D	**1.25**	1.62	2.02	-5.32
Corner E	-2.62	-2.21	-5.91	-7.38

Table V
COMPARISON AGAINST OTHER LEARNING-BASED METHODS

Methods	LPR	ITE	SL	RC	SMRT	VVCM
Salvaji et al. [40]	-	√	-	-	-	-
Spielberg et al. [41]	-	√	√	-	√	√
Evans et al. [42]	-	√	√	√	√	√
Ghignone et al. [43]	√	-	√	-	√	-
Proposed	√	√	√	√	√	√

Abbreviations: LPR: Local perception-based race; ITE: Improved training efficiency; SL: Shorter laptime; RC: Reduced collisions; SMRT: Simulation with multiple racetracks; VVCM: Visible variation of control commands; -: not considered or not given.

overlooks other key factors, including reducing laptime, fewer collisions, validating performance across multiple tracks, and providing visualizations of control commands. On the other hand, [41] introduces a DRL that encompasses improved training efficiency, shorter laptime, validation across various tracks, and clear visualization of control commands. However, it overlooks the aspect of reducing collisions. In contrast, the algorithms proposed in [42] considered all the factors in both [40] and [41], but still heavily relies on global perception. Furthermore, [43] focuses solely on local perception, emphasizing shorter laptime and validation across various tracks. However, [43] neglects improvements in training efficiency, collision reduction, and variations in control commands.

The proposed algorithm reduces dependency on sophisticated equipment and achieves enhanced training efficiency. Moreover, the laptime is reduced and collisions are avoided, thereby the overall racing performance is improved. Furthermore, validations on multiple tracks have been made, while interpretable control commands are provided, showcasing the generalization and interpretability of the proposed algorithm.

VII. DISCUSSION

The PPO-C algorithm typically surpasses comparative benchmarks by achieving greater training efficiency, higher average rewards, collision avoidance, and reduced laptime. Notably, while PPO-C approaches the highest training scores, it remains approximately 100 points behind, indicating the room for improvement. Future developments aim to narrow this gap, ideally to within 50 points of the top score. Although the PPO-C demonstrates proficiency in static environments, its performance in dynamic settings requires further validation. Additionally, there is potential to decrease laptime, as the PPO-C has not yet completely optimized for inner track navigation, as shown in Fig. 11. Before real-world application, the PPO-C's policy network and reward function must undergo refinement and rigorous testing to ensure safety and reliability. Moreover, prior to real-world implementation, a higher-fidelity simulation environment will be utilized to bridge the gap between simulation and actual conditions effectively.

VIII. CONCLUSION

This paper proposed a local perception-based, image-efficient, and balanced reward-orientated PPO-C for autonomous racing. The PPO-C aims to improve the training efficiency and driving performance of the racing car. To enhance the attention to critic steps, a balanced reward function is used to balance the historical and current rewards during the training. To enhance safety in exploration, a curiosity mechanism is introduced to focus on the dangerous racing periods. The results demonstrate that as training time increased, the proposed PPO-C improves its average scores with a higher degree of safety. Comparisons among the PPO-C and other three representative DRL algorithms were conducted, showing that the proposed algorithm outperforms in terms of no collision, shorter laptime, shorter training time, and higher average rewards. In the future, extensive research will be conducted in several aspects, including 1) verifying the racing ability of PPO-C under more uncertain conditions, 2) optimizing the racing process considering diverse objectives such as riding comfort, and 3) extending the algorithm to team competitions by using multiple agents.

REFERENCES

[1] J. Betz, A. Wischnewski *et al.*, "A software architecture for an autonomous racecar," in *Proceedings of the IEEE Vehicular Technology Conference*. IEEE, 2019, pp. 1–6.
[2] J. Betz, H. Zheng *et al.*, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.
[3] D. 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.
[4] C. Jung, A. Finazzi *et al.*, "An autonomous system for head-to-head race: Design, implementation and analysis; team KAIST at the indy autonomous challenge," *arXiv preprint arXiv:2303.09463*, 2023.
[5] A. Raji123 *et al.*, "er. autopilot 1.0: The full autonomous stack for oval racing at high speeds."
[6] J. Betz *et al.*, "Tum autonomous motorsport: An autonomous racing software for the indy autonomous challenge," *Journal of Field Robotics*, vol. 40, no. 4, pp. 783–809, 2023.
[7] J. Kabzan, M. I. Valls *et al.*, "Amz driverless: The full autonomous racing system," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1267–1294, 2020.

[8] J. Lu, L. Han *et al.*, "Event-triggered deep reinforcement learning using parallel control: A case study in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2821–2831, 2023.

[9] D. Pathak, P. Agrawal *et al.*, "Curiosity-driven exploration by self-supervised prediction," in *International Conference on Machine Learning*, 2017, pp. 2778–2787.

[10] M. Bevilacqua, A. Tsourdos, and A. Starr, "Particle swarm for path planning in a racing circuit simulation," in *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2017, pp. 1–6.

[11] S. Lovato and M. Massaro, "Three-dimensional fixed-trajectory approaches to the minimum-lap time of road vehicles," *Vehicle System Dynamics*, vol. 60, no. 11, pp. 3650–3667, 2022.

[12] S. Grollius, M. Ligges, J. Ruskowski, and A. Grabmaier, "Concept of an automotive lidar target simulator for direct time-of-flight lidar," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 825–835, 2021.

[13] C. You and P. Tsiotras, "High-speed cornering for autonomous off-road rally racing," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 485–501, 2021.

[14] F. Sauerbeck, L. Baierlein *et al.*, "A combined lidar-camera localization for autonomous race cars," *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 12-05-01-0006, pp. 61–71, 2022.

[15] F. Massa *et al.*, "Lidar-based gnss denied localization for autonomous racing cars," *Sensors*, vol. 20, no. 14, p. 3992, 2020.

[16] F. Sauerbeck, S. Huch *et al.*, "Learn to see fast: Lessons learned from autonomous racing on how to develop perception systems," *IEEE Access*, vol. 11, pp. 44 034–44 050, 2023.

[17] K. Huang, B. Shi *et al.*, "Multi-modal sensor fusion for auto driving perception: A survey," *arXiv preprint arXiv:2202.02703*, 2022.

[18] L. Hewing, A. Liniger *et al.*, "Cautious NMPC with gaussian process dynamics for autonomous miniature race cars," in *Proceedings of the European Control Conference*, 2018, pp. 1341–1348.

[19] P. A. Theodosis and J. C. Gerdes, "Nonlinear optimization of a racing line for an autonomous racecar using professional driving techniques," in *Dynamic Systems and Control Conference*, vol. 45295. American Society of Mechanical Engineers, 2012, pp. 235–241.

[20] J. L. Vázquez *et al.*, "Optimization-based hierarchical motion planning for autonomous racing," in *IEEE conference on intelligent robots and systems*. IEEE, 2020, pp. 2397–2403.

[21] Y. Song, H. Lin *et al.*, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 9403–9409.

[22] P. R. Wurman, S. Barrett *et al.*, "Outracing champion gran turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[23] F. Fuchs *et al.*, "Super-human performance in gran turismo sport using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.

[24] G. Basile, A. Petrillo *et al.*, "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.

[25] M. A. Hebaish, A. Hussein *et al.*, "Towards safe and efficient modular path planning using twin delayed DDPG," in *Proceedings of the IEEE Vehicular Technology Conference*, 2022, pp. 1–7.

[26] A. Remonda, S. Krebs *et al.*, "Formula rl: Deep reinforcement learning for autonomous racing using telemetry data," *arXiv preprint arXiv:2104.11106*, 2021.

[27] J. Niu, Y. Hu *et al.*, "Two-stage safe reinforcement learning for high-speed autonomous racing," in *2020 IEEE International Conference on Systems, Man, and Cybernetics*, 2020, pp. 3934–3941.

[28] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[29] F. Tong, R. Liu *et al.*, "Multi-policy soft actor-critic reinforcement learning for autonomous racing," in *2024 IEEE 18th International Conference on Advanced Motion Control (AMC)*, 2024, pp. 1–7.

[30] S. Siboo, A. Bhattacharyya *et al.*, "An empirical study of ddpg and ppo-based reinforcement learning algorithms for autonomous driving," *IEEE Access*, vol. 11, pp. 125 094–125 108, 2023.

[31] L. Wang, Q. Cai, Z. Yang, and Z. Wang, "Neural policy gradient methods: Global optimality and rates of convergence," *arXiv preprint arXiv:1909.01150*, 2019.

[32] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International conference on machine learning*. PMLR, 2019, pp. 2052–2062.

[33] F. Ye, X. Cheng *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.

[34] C. Qi, C. Wu *et al.*, "UAV path planning based on the improved ppo algorithm," in *2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE)*, 2022, pp. 193–199.

[35] W. Chen, K. K. L. Wong, S. Long, and Z. Sun, "Relative entropy of correct proximal policy optimization algorithms with modified penalty factor in complex environment," *Entropy*, vol. 24, no. 4, p. 440, 2022.

[36] J. Bharadiya, "Convolutional neural networks for image classification," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 5, pp. 673–677, 2023.

[37] F. N. Iandola, S. Han *et al.*, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[38] I. Parberry, *Introduction to Game Physics with Box2D*. CRC Press, 2017.

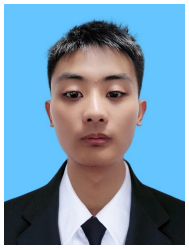
[39] M. Estrada *et al.*, "Feedback linearization of car dynamics for racing via reinforcement learning," *arXiv preprint arXiv:2110.10441*, 2021.

[40] A. Salvaji, H. Taylor, *et al.*, "Racing towards reinforcement learning based control of an autonomous formula sae car," *arXiv preprint arXiv:2308.13088*, 2023.

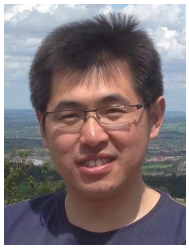
[41] N. A. Spielberg, M. Templer, *et al.*, "Learning policies for automated racing using vehicle model gradients," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 130–142, 2023.

[42] B. D. Evans *et al.*, "Safe reinforcement learning for high-speed autonomous racing," *Cognitive Robotics*, vol. 3, pp. 107–126, 2023.

[43] E. Ghignone, N. Baumann *et al.*, "Tc-driver: A trajectory conditioned reinforcement learning approach to zero-shot autonomous racing," *Field Robotics*, vol. 3, no. 1, pp. 637–651, 2023.



Zhen Tian received the B.Eng. degree in Electronic and Electrical engineering from University of Strathclyde, Glasgow, UK, in 2020. He is currently pursuing the Ph.D. degree with the James Watt School of Engineering, University of Glasgow, UK. His research interests include safe decision making in autonomous driving, deep reinforcement learning and control engineering.



Dezong Zhao received the B.Eng. and M.S. degrees from Shandong University, Jinan, China, in 2003 and 2006, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2010, all in Control Science and Engineering. He is a Reader in Autonomous Systems with the James Watt School of Engineering, University of Glasgow and a Turing Fellow with the Alan Turing Institute. He was awarded a Royal Society-Newton Advanced Fellow in 2020 and an EPSRC Innovation Fellow in 2018.

Zhihao Lin received the M.S. degree from the College of Electronic Science and Engineering, Jilin University, Changchun, China. He is currently pursuing the Ph.D. degree with the James Watt School of Engineering, University of Glasgow, UK. His research interests focus on multi-sensor fusion SLAM systems and robot perception in complex scenarios.

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3432713

Wenjing Zhao received the Ph.D. degree in Traffic Engineering with Central South University, Changsha, China, in 2022. She is currently a Postdoctoral Fellow with the Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China. Her research interests include traffic safety, driving behaviour analysis, and connected vehicles.

David Flynn received the B.Eng. degree (Hons.) in Electrical and Electronic engineering, the M.Sc. degree (Distinction) in Microsystems, and the Ph.D. degree in Microscale Magnetic Components from Heriot-Watt University, Edinburgh, UK, in 2002, 2003, and 2007, respectively. He is a Professor of Cyber Physical Systems at University of Glasgow. He is a co-founder of the UK's EPSRC National Centre for Energy System Integration and the UK Offshore Robotics and Artificial Intelligence Hub for Offshore Energy Asset Integrity Management.

Yuande Jiang received the B.S. degree in Automobile Application Engineering from Chang'an University, Xi'an, China, in 2014, and the Ph.D. degree in Vehicle Engineering from Jilin University, Changchun, China, in 2019. He is currently a Lecturer with the School of Information Engineering, Chang'an University. His research interests include autonomous vehicle decision-making algorithm and autonomous vehicle testing.

Daxin Tian is currently a Professor in the School of Transportation Science and Engineering, Beihang University, Beijing, China. He is IEEE Intelligent Transportation Systems Society Member, and IEEE Vehicular Technology Society Member, etc. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligent systems.

Yuanjian zhang received the M.S. degree from Coventry University, Coventry, U.K., in 2013, and the Ph.D. degree from Jilin University, Changchun, China, in 2018, both in Automotive Engineering. He is a Professor at College of Automotive Studies, Tongji University. His research interests include advanced control of electric vehicle powertrains, vehicle-environment-driver cooperative control, vehicle dynamic control and intelligent control for driving assistance systems.

Yao Sun received the B.S. degree in Mathematical Science, and the Ph.D. degree (Honors) in Communication and Information System from University of Electronic Science and Technology of China in 2014 and 2019, respectively. He is a Lecturer with the James Watt School of Engineering at University of Glasgow, UK. His research interests include intelligent wireless networking, network slicing, blockchain system, internet of things and resource management in mobile networks.