2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Physics-Constrained Imitation Learning for Autonomous Racing

Haohan Yang†, Haochen Liu†, Yanxin Zhou, Shuge Wu, and Chen Lv*

***Abstract*— Autonomous racing has become increasingly popular in both academia and industry as a testbed for pushing general autonomous driving modules, such as perception, planning, and control, to their limits. Although traditional control approaches can generate optimal control sequences at the edge of the racing vehicles' physical controllability, they are highly sensitive to the accuracy of modeling parameters, such as tire model coefficients. Meanwhile, end-to-end learning methods are susceptible to distributional shifts, leading to unpredictable and irreversible failures. To address these challenges, this work introduces a physics-constrained imitation learning (PCIL) framework that effectively leverages the advantages of deep learning techniques and knowledge-driven strategies. Specifically, a fallback strategy would be automatically triggered when the vehicle states exceed predefined physical constraints. Meanwhile, the data from the knowledge-driven strategy will be augmented into the original dataset, and repeated re-training using an aggregated dataset could progressively improve PCIL. A series of simulations and real-world shadow testing are conducted at the Yas Marina circuit, and experimental results demonstrate superior performance compared to state-of-the-art methods, which suggests that it provides a promising solution for real-world autonomous racing.**

I. Introduction

In recent years, autonomous driving has made significant progress across different modules, including perception, prediction [1], behavior planning [2], etc. Nevertheless, it has not yet gained widespread acceptance among the public due to safety concerns. In autonomous racing, the vehicles are required to operate within their dynamic limits, which can demonstrate the ultimate capabilities of autonomous driving systems [3], [4]. Accordingly, superior performance in autonomous racing would foster public trust in autonomous vehicles, especially in high-speed, low-interactive scenarios.

Autonomous racing is characterized by high speeds that render conventional algorithms for different modules ineffective [5]. For instance, racing vehicles are susceptible to instability when the control module does not incorporate dynamic model-based state estimation [6]. However, real-world racing facilities are prohibitively expensive, limiting extensive parameter tuning of conventional strategies. Due to the static environment of racetracks, learning-based strategies merely require a few expert demonstrations for training. Nonetheless, imbalanced data distribution could degrade network model performance in long-tail racetrack sections, resulting in unpredictable and irreversible collisions.

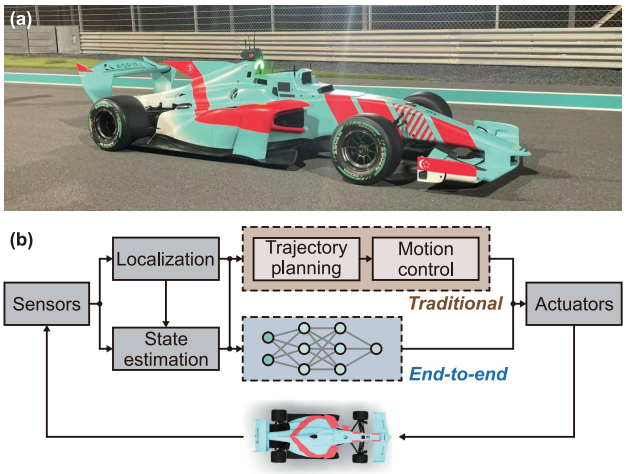

Fig. 1. Overview of autonomous racing systems. (a) An autonomous Super Formula racing vehicle in single-car time trials. (b) Two categories of autonomous racing software stacks, i.e., traditional and end-to-end.

To address the aforementioned challenges, we propose a **P**hysics-**C**onstrained **I**mitation **L**earning (PCIL) approach for autonomous racing. Our developed approach aims to enhance the model's vehicle handling capabilities by acquiring high-quality expert demonstrations while preventing inappropriate maneuvers that may lead to loss of control. Specifically, a conservative fallback strategy can be activated when the racing vehicle's physical state exceeds the predefined threshold. Therefore, PCIL seamlessly integrates the strengths of knowledge-driven and learning-based methods regarding efficiency, adaptability, safety, etc. Notably, this work focuses on single-car time trials to evaluate PCIL's vehicle handling limits, as shown in Fig. 1a. The paradigm of PCIL could be smoothly extended to other competitive tasks, such as multi-car racing, by incorporating additional input information. Our contributions can be summarized as follows:

- A physics-constrained learning framework is proposed to leverage the advantages of learning-based algorithms while ensuring safety by physical constraint-triggered fallback mechanisms, which also suggests a viable generic technique for other extreme-condition tasks.
- A knowledge-driven strategy is designed as a fallback strategy, and its takeover data would be augmented into the original dataset. Furthermore, repeated re-training can progressively enhance the PCIL agent's performance.
- Extensive simulation tests and experiments are carried out to validate the effectiveness of the PCIL, and the results demonstrate that our proposed method outperforms other

This work was supported in part by K2, Abu Dhabi, UAE, under Research Collaboration Agreement (No. REQ0474834), the Agency for Science, Technology and Research (A*STAR), Singapore, under the MTC Individual Research Grant (M22K2c0079), and the Ministry of Education (MOE), Singapore, under the Tier 2 Grant (MOE-T2EP50222-0002).

The authors are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 639798.
† These authors contributed equally.
* Corresponding author: Chen Lv (lyuchen@ntu.edu.sg).

state-of-the-art methods in terms of lap time, safety, etc.

The rest of this paper is structured as follows: Section II reviews related work in end-to-end autonomous driving and autonomous racing methods. Section III details the architecture of the proposed PCIL framework. Section IV presents the experimental setup and results, showing the effectiveness of PCIL compared to other approaches. Finally, Section V concludes the paper and outlines future work.

II. Related Work

A. End-to-end Autonomous Driving

End-to-end autonomous driving can be broadly categorized into two groups. One is modular end-to-end pipelines, wherein the hierarchical architecture enables independent data-driven optimization of each module [7]. Although this method has improved interpretability and reliability, its error propagation renders it unsuitable for autonomous racing. The other involves directly mapping raw sensor data to control commands, which is cost-efficient and adaptable to complex environments. Such a technique can be accomplished by reinforcement learning (RL) [8] and imitation learning (IL) [9]. RL adopts an exploratory trial-and-error manner, allowing it to independently optimize its policy by interacting with the controlled environment. Many studies have demonstrated its effectiveness in specific scenarios, such as city intersections [10], ramp merging [11], etc. However, meticulously designed rewards generally are required to achieve superior performance. In addition, RL models inevitably suffer from the simulation-to-reality (sim-to-real) gap since their training relies on self-explored data within the simulation [12]. In comparison, IL is a widely used data-driven method that effectively learns from collected human demonstration data. Especially, it can achieve satisfactory performance with minimal data in static driving scenarios. It is noted that IL models' performance depends on the distribution of collected data, which implies that the model may fail to deal with specific road segments with limited data. Dataset aggregation (DAgger) is a promising solution where the expert strategy can intervene at any time, with the collected data augmented into the training set [13]. Iterative training with augmented data could continually reduce intervention rates and enhance model performance.

B. Autonomous Racing

The motion control module is responsible for outputting commands to actuators, critically influencing autonomous racing performance [3], [4]. In conventional software stacks, different control strategies are developed to compute control commands based on the planned trajectory [5], [14], as shown in Fig. 1b. In [15], a tube-based model predictive control (MPC) approach was proposed to approximate a tube of reachable sets over the prediction horizon, thereby ensuring constraint satisfaction. Furthermore, some learning techniques have been integrated to enhance MPC strategies [16], [17], [18]. For example, an IL approach was employed to optimize the constraints of the safety filter, ensuring racing vehicle safety with respect to track boundaries when paired with any potentially unsafe control signal [18]. However, these methods still necessitate extensive parameter tuning in real-world applications.

Numerous studies explore data-driven methods for directly generating control commands, where model parameters can be optimized using the collected data [19], [20], [21], [22], as presented in Fig. 1b. An imitative RL method has been proposed to map visual information directly to control commands [19]. Although some studies have achieved racing tasks using visual inputs, this is not an optimal choice for single-car time trials, since essential information about the fixed racetrack, such as its boundaries, can be fully obtained offline [23]. In [21], a trajectory-conditioned RL approach was developed to reduce the effects of inaccuracies in tire parameter modeling. However, these studies focused merely sim-to-real transfer on the physical F1Tenth system, with vehicle speeds below 3 m/s. Real-world autonomous racing features higher speeds and more complex nonlinear dynamics, making the performance of these methods on such vehicles uncertain. On the whole, autonomous racing imposes two features on network models, i.e., high frequency and limited generalization. Specifically, a higher inference frequency suggests that smaller models are more suitable, and the static racetrack places minimal requirements on model generalization.

III. Methodology

In this section, we provide a detailed description of the proposed PCIL framework. The problem of learning-based autonomous racing is formalized firstly, then the architecture of our model and its training procedure are introduced, respectively.

A. Problem Formulation

PCIL could map vehicle states $\mathbf{s}$ and racetrack-related information $\mathbf{M}$ to the control command $\mathbf{a}$, as shown in Fig. 2. At each timestep t, the input state consists of three components,

$$\mathbf{s}_t = \left[\mathbf{s}_t^{\mathrm{pos}}, \mathbf{s}_t^{\mathrm{veh}}, \mathbf{s}_t^{\mathrm{tire}}\right] \tag{1}$$

where $\mathbf{s}^{\mathrm{pos}} = [x, y, z, \psi, \alpha, \phi]$ is vehicle's position and pose information, including global coordinates, yaw, pitch, and roll angles. $\mathbf{s}^{\mathrm{veh}} = [v_x, v_y, v_z, w_{lf}, w_{lr}, w_{rf}, w_{rr}]$ is the vehicle's kinematic state, including vehicle's three-dimensional velocities and the four wheels' angular speeds. $\mathbf{s}^{\mathrm{tire}} \in \mathbb{R}^8$ denotes pressures and temperatures of four tires. Also, the racetrack-related information $\mathbf{M}$ contains road boundaries and the optimal trajectory extracted from offline global planning,

$$\mathbf{M} = \left[\mathbf{M}^{\mathrm{lb}}, \mathbf{M}^{\mathrm{rb}}, \mathbf{M}^{\mathrm{opt}}\right] \tag{2}$$

where $\mathbf{M}^{\mathrm{lb}} = [m_x, m_y, m_z, m_\kappa, m_\tau]$ is the left boundary's spatial location, curvature, and torsion, respectively, and $\mathbf{M}^{\mathrm{rb}}$ is the same information of the right boundary. $\mathbf{M}^{\mathrm{opt}}$ additionally contains target speeds. To reduce the computational cost, this work considers only the next H time horizons of

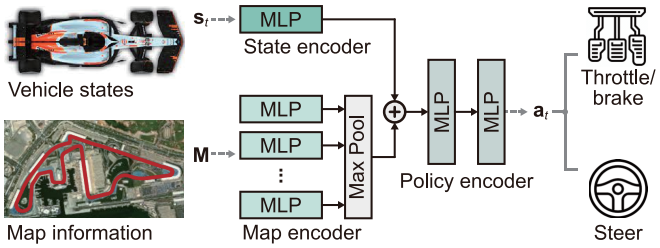

Fig. 2. Overview of the developed PCIL policy for autonomous racing.

map information. The PCIL model outputs a command $\mathbf{a}_t$ involving both longitudinal and lateral control,

$$\mathbf{a}_t = \left[a_t^{\text{lon}}, a_t^{\text{lat}}\right] \tag{3}$$

where a^{lon} is the throttle/brake and a^{lat} is the steering angle.

This work aims to obtain a PCIL policy π with parameters θ, who can output most probable control commands $\mathbf{a}$ based on the information of vehicle states

$$\max_{\theta} \log \pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{M}) \tag{4}$$

B. Physics-Constrained Imitative Racing Agent

To imitate expert-level racing agent, PCIL establishes a swift learning-based framework to deliver racer-like and agile control under constrained dynamical limits. Specifically, the imitation agent $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{M})$ is formulated by the lightweight encoder-decoder structure, with linear complexity of multi-layer perceptrons (MLPs) to ensure efficient responses.

Scene Encoder. Separated encoders are devised to effectively extract the feature representations from estimated racing vehicle's status $\mathbf{s}_t$ and observational prior $\mathbf{M}$ from racetrack. Racing vehicle state is directly mapped as $\mathbf{H}_s = \text{MLP}(\mathbf{s}_t)$. Also, we leverage a PointNet-like encoder [24] to extract racetrack sequence features by incorporating local priors, $\mathbf{H}_\mathbf{M} = \max_i(\text{MLP}(\mathbf{m}_i) + \max_i(\text{MLP}(\mathbf{m}_i))), \mathbf{m}_i \in \mathbf{M}$. Specifically, the racetrack encoder explicitly incorporates features for racetrack boundaries and forward reference information provided by the collected expert demonstrations. The encoded features are then combined as $\mathbf{H}$ for action decoding.

Agent Decoder. To ensure the imitation of expert racing demonstrations while maintaining physical feasibility, we introduce a physical dynamic model that incorporates explicit dynamics to enforce physical constraints. First, the control actions are directly decoded as $\hat{\mathbf{a}}_t$, and both throttle/brake and steering commands are normalized to [–1, 1].

A differentiable vehicle dynamics model is employed to verify PCIL's compliance with physical constraints [25]. In this study, only horizontal constraints, including longitudinal/lateral velocities and longitudinal/lateral/yaw accelerations, are selected to enhance computational efficiency,

$$\hat{\mathbf{q}} = \begin{cases} \dot{x} = v_x \cos(\psi) - v_y \sin(\psi) \\ \dot{y} = v_x \sin(\psi) + v_y \cos(\psi) \\ \dot{v}_x = (F_x - F_l)/m \\ \dot{v}_y = -(F_{yf} \cos\psi + F_{yr})/m - v_x \dot{\psi} \\ \ddot{\psi} = (-l_f F_{yf} \cos\psi + l_r F_{yr})/I_z \end{cases} \tag{5}$$

where m and I_z are the racing vehicle's mass and yaw inertia, respectively. l_f and l_r are the distance from the center of gravity (CoG) to the front and rear wheels, respectively. F_x is the longitudinal force, F_{yf} and F_{yr} are the lateral forces of the front and rear wheels, respectively. The calculation of these forces follows the Magic Formula structure [26]. F_l denotes the sum of racing vehicle's aerodynamics drag and rolling resistance,

$$F_l = \frac{1}{2}\rho C_d A_F v_x^2 + C_r m g \tag{6}$$

wherein C_d and C_r are the coefficients of drag and rolling resistance, respectively. A_F denotes the vehicle's frontal area, and ρ is the air density. g is the gravitational acceleration.

Output actions $\hat{\mathbf{a}}_t$ can then be directly guided by the predicted dynamic states $\hat{\mathbf{q}}_{t+1}$ based on the racing vehicle's dynamic model, and any state in $\mathbf{q}$ exceeds its limit would trigger the fallback strategy. It should be noted that precise dynamic parameters are unnecessary since we can set a relatively loose threshold in practice. With the same input states, any stable controller can be incorporated into the PCIL agent as a fallback strategy. A tube-MPC algorithm is employed [15], denoted as π_f, in this work. It primarily aims to maintain vehicle stability rather than push dynamic limits, thereby eliminating fine-tuned parameter adjustments. To prevent abrupt command changes during policy switching, a gradually varying weight allocation is used to encourage a smooth transition between π_θ and π_f. To ensure absolute safety in real-world racing, we also design a safe stop strategy π_safe based on displacement deviation, localization reliability, etc. to avoid irreversible collisions. This case falls beyond the scope of this paper and is therefore omitted here. Accordingly, the deployed PCIL policy is formulated as

$$\mathbf{a}_t = \begin{cases} \pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{M}), \mathbf{s}_t \in \mathcal{S}_\mathcal{N} \\ \pi_\text{f}(\hat{\mathbf{a}}_{t-1}, \mathbf{s}_t, \mathbf{M}), \mathbf{s}_t \notin (\mathcal{S}_\mathcal{N} \cup \mathcal{S}_\mathcal{R}) \\ \pi_\text{safe}(\mathbf{s}_t, \mathbf{M}), \mathbf{s}_t \in \mathcal{S}_\mathcal{R} \end{cases} \tag{7}$$

wherein $\mathcal{S}_\mathcal{N}$ and $\mathcal{S}_\mathcal{R}$ represent normal and risky states of the racing vehicle. Furthermore, data generated by fallback strategies are collected and incorporated into the original dataset for agent re-training. This setup progressively reduces the frequency at which the agent triggers fallback strategies.

C. PCIL Agent Training

The high-frequency control of racing vehicles exacerbates the misalignment between logged expert actions and executed controls, leading to increased covariance shifts during deployment. By constraining states to the dynamic limits of the racing vehicle, PCIL further improves physical feasibility. Its objective is formulated as,

$$\mathcal{L}(\theta) = \mathcal{L}_\text{NLL} + \lambda \cdot \mathbf{1}_{\{\mathbf{s} \notin (\mathcal{S}_\mathcal{N} \cup \mathcal{S}_\mathcal{R})\}} \sum \mathcal{L}_h(\hat{\mathbf{q}}, \mathbf{q}) \tag{8}$$

where the $\mathcal{L}_\text{NLL}$ represents negative log-likelihood loss that is formulated as

$$\mathcal{L}_\text{NLL} = \mathbb{E}_{(\mathbf{s},\mathbf{M},\mathbf{a})\sim\mathcal{D}} \left[\frac{\log \hat{\sigma}_\theta^2(\mathbf{a})}{2} + \frac{(\mathbf{a} - \mu_\theta(\mathbf{s}, \mathbf{M}))^2}{2\hat{\sigma}_\theta^2(\mathbf{s}, \mathbf{M})} \right] \tag{9}$$

Algorithm 1: Physics-constrained Imitation Learning Framework

Input: training set $\mathcal{D}$, mini-batch size B, regularization coefficient λ

Stage 1: obtain PCIL policy π_θ

for *training step* $\leftarrow 1$ **to** *max iteration* **do**

  Sample N samples from training set $\mathcal{D}$;

  Calculate policy objective $\mathcal{L}(\theta)$ by Eq. (8);

  Update policy network parameter θ;

end

Stage 2: optimize PCIL policy π_θ

Collect data from racetrack segments where PCIL breaches constraints;

Augment the collected data into training set $\mathcal{D}$;

Repeat Stage 1 for policy re-training;

TABLE I

MODEL STRUCTURE AND TRAINING PARAMETERS

Parameters	Value	Description
time horizon	20	future horizon of map information
dropout rate	0.1	fraction of neurons randomly deactivated
activation	ReLU	activation functions for network layers
learning rate	0.001	initial learning rate of the network
optimizer	Adam	adaptive model update algorithm
mini-batch size	512	number of training samples in one batch
epoch	30	maximum training iterations
loss weight (λ)	0.5	regularization coefficient of loss function

Also, $\mathbf{1}$ is an indicator function triggered only when the PCIL agent violates physical constraints. Therefore, this term penalizes racing vehicle states that exceed physical limits using smooth-L1 loss,

$$\mathcal{L}_h = \|\hat{\mathbf{q}} - \mathbf{q}\|_1 \tag{10}$$

This joint loss function design would reinforce state awareness during regular IL. Detailed training parameters are listed in Table I, and the pseudo-code of our model's learning strategy is provided in Algorithm 1.

IV. EXPERIMENTAL SETTINGS

The experimental validation is conducted in simulation and real-world environments, respectively. In this section, we provide detailed descriptions of dataset construction, baseline methods, and evaluation metrics.

A. Dataset Construction

In this study, the models are primarily evaluated on the Fédération Internationale de l'Automobile (FIA) Grade 1 circuit at Yas Marina in Abu Dhabi. This circuit layout features 16 turns and covers a distance of 5.281 km. Training data is collected from both simulations and real-world environments, respectively, at a frequency of 100 Hz. The details are listed below,

Simulations. A high-fidelity autonomous racing simulator, specifically Assetto Corsa (AC), is employed for its accurate physics engine and realistic vehicle dynamics. A total of 20 laps of expert data are collected in the AC simulator. Their lap times are similar, with the fastest lap representing the optimal trajectory under given conditions.

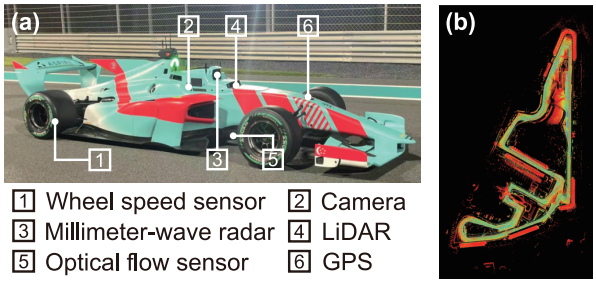

Fig. 3. Platform for autonomous racing data collection. (a) Racing vehicle equipped with different sensors. (b) Yas Marina circuit.

Shadow Testing. Real-world data are collected based on a Super Formula 23 (SF23) vehicle, as shown in Fig. 3. A multi-sensor fusion-based localization scheme is employed to ensure the validity of data collected at high speeds [27]. Based on offline global planning, 5 laps of expert data are collected.

B. Baselines

A series of learning-based approaches are implemented as baselines to benchmark the performance of our proposed approach. A detailed description of baseline methods is listed below,

1) TAL-TD3: a trajectory-aided learning approach with the twin delayed deep deterministic policy gradient (TD3) algorithm [20]. This baseline is employed for comparatively evaluating the significance of IL techniques.

2) TAL-SAC: a trajectory-aided learning approach with the soft actor-critic (SAC) algorithm. SAC is generally more forgiving to hyperparameter choices compared with TD3, thus it is more suitable for real-world applications.

3) IQL: an implicit Q learning algorithm, which is an offline RL method [28]. This baseline is employed for comparatively assessing the significance of IL.

4) TD3-BC: an offline RL algorithm that integrates TD3 with behavior cloning (BC) [29]. The BC term could constrain the policy toward the training dataset. This baseline leverage the advantages of both IL and RL.

5) IL: a vanilla imitation learning method. This baseline is designed for comparatively evaluating the significance of physical constraints.

To achieve a fair comparison, all baseline methods have uniform state and action spaces, and they also share the same model structure. The optimal expert trajectory is employed as a reference to guide the online exploration of RL-based policies. Offline RL-based and IL-based methods share the same expert demonstrations for model training.

C. Evaluation Metrics

Different metrics are selected to evaluate model performance from various aspects, and the details are as follows:

1) Lap time – overall efficiency optimization. A shorter lap time indicates better comprehensive performance, including racing line optimization, vehicle dynamics control, etc.

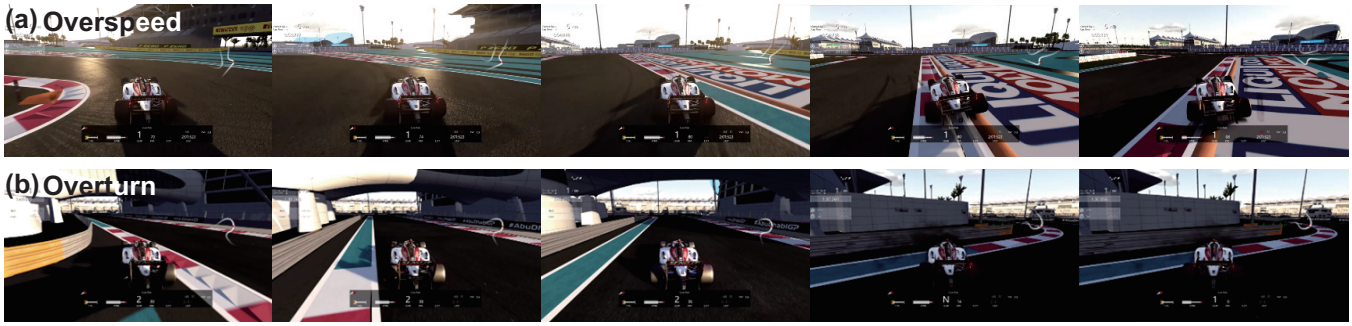

Fig. 4. Representative accident cases in autonomous racing. (a) Insufficient deceleration before corner entry. (b) Delayed steering adjustment after corner exit.

2) Maximum speed – utilization of vehicle dynamics and stability control. High maximum speed suggests effective acceleration management and high-speed stability of models.

3) Success rate – robustness and reliability. A higher success rate reflects better adaptability to environmental variations and robustness against disturbances.

It is noted that the lap time and maximum speed are excluded from the analysis if an irreversible collision occurs in this lap.

V. Results and Discussions

In this section, we first comprehensively evaluate the proposed and baseline autonomous racing approaches in simulations. Furthermore, shadow testing is conducted on the proposed PCIL strategy using real-world collected data, aiming to validate the model's continuous performance improvement after acquiring augmented data.

A. Simulations

All trained models are deployed in the AC simulator for evaluation. To mitigate the influence of aleatoric uncertainty on performance evaluation, we conduct 20 single-car time trials for each approach, and the test results are listed in Table II. It can be observed that most methods struggle to completely avoid collisions. Two primary types of accidents commonly occur, i.e., overspeed and overturn, as shown in Fig. 4. Specifically, overspeed results from insufficient deceleration during cornering, leading to curb impact or racetrack departure. Overturn stems from improper steering control during cornering, as shown in Fig. 4b. RL-based methods show potential for autonomous racing, such as TAL-SAC achieving the highest speed. However, it still tends to overspeed since the rarity of hairpin turns requiring speeds below 40 km/h hinders its effective exploration. Although RL-based algorithms may be further improved by optimizing the reward design, they require the time-consuming manual design of separate reward functions for different racetrack structures. Since the training data are all of high quality, the advantages of offline RL are not fully demonstrated in the static racing scenario. IL learns from expert demonstrations more efficiently, thereby achieving competitive lap times and maximum speeds. However, its success rate remains low due to sparse data in certain track regions. In comparison,

TABLE II

Performance comparison across different methods.

Method	Lap time (s)	Maximum speed (km/h)	Success rate
Expert	**106.21±1.67**	283.49	**1.00**
TAL-TD3	128.12±6.22	242.91	0.75
TAL-SAC	110.18±2.25	**283.61**	0.90
IQL	115.41±5.33	270.19	0.65
TD3-BC	111.73±4.63	273.55	0.80
IL	107.82±3.17	280.04	0.30
PCIL (ours)	109.82±1.13	277.93	**1.00**

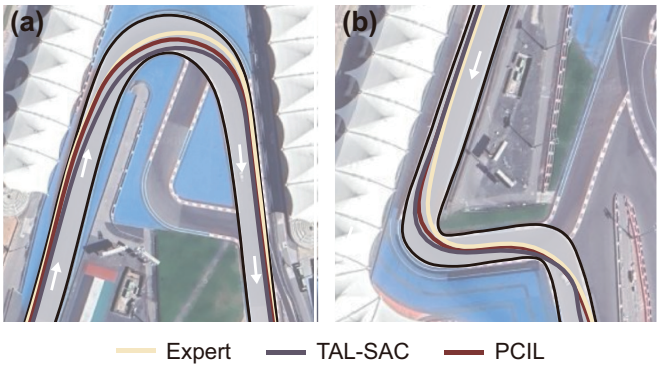

Fig. 5. Trajectories of representative methods on typical corners. (a) Compound corner. (b) Hairpin.

the proposed PCIL agent prevents irreversible collisions by triggering a fallback strategy, albeit at the cost of a slight speed reduction.

We further visualized the trajectories of the representative methods in the compound corner and hairpin, respectively, as shown in Fig. 5. It can be observed that the expert trajectory tends to enter the compound corner from the outside to maximize the corner radius, allowing the maintenance of higher speeds throughout the turn. In contrast, the entry speed is sacrificed to better utilize the full track width on exit for quicker acceleration in the hairpin. However, TAL-SAC does not explore racing lines closer to the vehicle's dynamic limits. A fundamental dilemma in applying online RL-based algorithms to autonomous racing is that the absence of a reference trajectory hinders effective data exploitation for

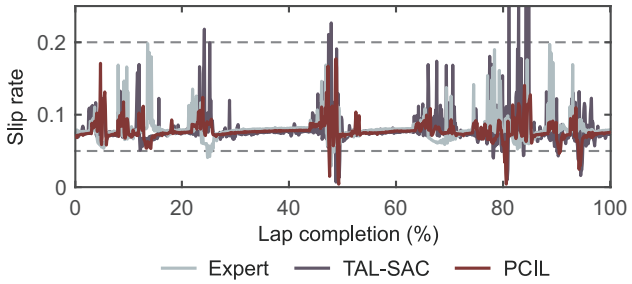

Fig. 6. Slip rates of representative methods over a single racing lap.

TABLE III
PERFORMANCE COMPARISON UNDER DIFFERENT WEIGHTS OF PHYSICAL CONSTRAINTS.

Weight (λ)	Lap time (s)	Maximum speed (km/h)	Success rate
0	**107.82±3.17**	**280.04**	0.30
0.2	109.15±1.64	278.62	0.85
0.5	109.82±1.13	277.93	**1.00**
1.0	111.07±1.09	277.89	**1.00**

policy training, whereas its presence inherently constrains exploration. In comparison, the proposed PCIL's trajectory more closely aligned with that of the expert. Due to the simplified dynamics model adopted in this study, additional physical constraints prevent PCIL from fully matching the expert's performance.

Slip rate measures the relative difference between wheel and racing vehicle speeds, reflecting the model's capability in handling the racing vehicle at its dynamic limits. It is expected to maintain the slip ratio within 0.1–0.2, thereby achieving high traction/braking forces while preventing wheel slip. As shown in Fig. 6, the slip rate of expert demonstrations is generally maintained between 0.05 and 0.20, frequently reaching 0.15–0.20 during cornering maneuvers. Accordingly, the expert driver enables the racing vehicle to maintain high lateral forces while preventing excessive oversteer or understeer. We further selected representative methods, i.e., TAL-SAC (RL-based), and our proposed PCIL (IL-based), for further analysis. Obviously, the TAL-SAC agent's slip rate exceeds 0.20 multiple times. This phenomenon indicates that the racing vehicle experiences slipping, which would be highly dangerous in real-world applications. In comparison, the PCIL agent also effectively exploits the tire's optimal traction limit.

To investigate the influence of the designed physical constraint, an ablation study is performed on its weighting coefficient λ. As presented in Table III, the PCIL degenerates to the vanilla IL that fully exploits expert demonstrations when the weight is set to 0. However, it becomes highly susceptible to distributional shift, resulting in accidents under out-of-distribution conditions. As the weight increases, the PCIL progressively refrains from operating in near-limit driving conditions. The occurrence of accidents can be effectively mitigated when the weight is set to 0.5 or 1.0. In this work,

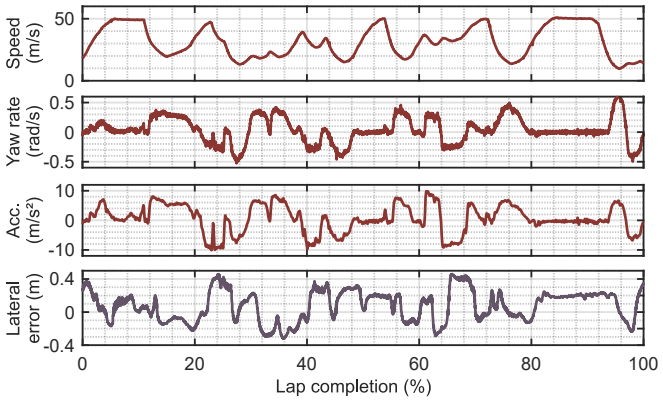

Fig. 7. Representative states of the racing vehicle during field testing, including speed, yaw rate, lateral acceleration, and lateral error.

λ=0.5 represents an optimal balance between exploiting the dynamic limits and maintaining a high success rate. With a more precise dynamics model, λ could be further reduced while maintaining collision-free performance. Due to some unmodeled factors, such as tire wear, the PCIL exhibits slightly lower performance than the expert driver. Nevertheless, the proposed PCIL has the potential to effectively leverage the strengths of different expert drivers by jointly learning from their demonstrations.

B. Shadow Testing

We first implement the traditional modular pipeline on the autonomous racing vehicle, and its operational states are shown in Fig. 7. The maneuvers, such as curb-riding, are prohibited in practice due to safety concerns. Nevertheless, the racing vehicle achieves a maximum speed of 180 km/h and yields a lap time of approximately 160 s. Moreover, the yaw rate and lateral acceleration exceed 0.4 $\mathrm{rad/s}$ and 9 $\mathrm{m/s^2}$, respectively, in certain corners, indicating that the racing vehicle is operating near its dynamic limits. To enable real-time performance evaluation while avoiding potential crashes or unsafe maneuvers, we employ shadow testing to assess the PCIL agent's capability for progressive performance improvement. As shown in Fig. 7, lateral errors relative to the planned global optimal trajectory consistently remains below 0.4 m, satisfying the prerequisites for shadow testing.

By learning from an augmented dataset, the PCIL agent's performance can continuously improve in practical racing. The positions where the PCIL agents encounter physical constraints within a single lap are illustrated in Fig. 8. It can be observed that the agent initially triggers the fallback strategy multiple times, especially at cornering positions. Variations in corner characteristics, such as curvature and width, at Yas Marina circuit result in low-frequency data in the training set. Consequently, the model exhibits limited performance in these racetrack segments. After augmenting data from these areas into the original dataset and re-training the model, the PCIL agent yields a significant performance improvement. Based on our real-world collected data, the model's violation rate of physical constraints decreased from

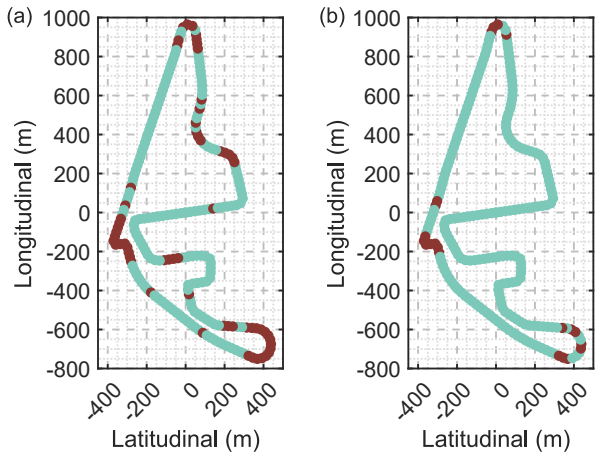

Fig. 8. Shadow testing results of the PCIL agent on real-world collected data. Green points indicate no fallback strategy is triggered, while red points indicate fallback activation. (a) Initial policy. (b) Optimized policy.

26.82% to 9.55%. With further training on additional data, the model is expected to eventually operate without triggering the fallback strategy.

VI. Conclusion

In this paper, we develop a physics-constrained imitation learning framework for autonomous racing. Specifically, a fallback strategy would take over the racing vehicle when its physical state exceeds the predefined constraints. Therefore, the PCIL is capable of integrating the strengths of knowledge-driven and learning-based methods in terms of efficiency, adaptability, safety, etc. Experimental results of both simulation and real-world tests demonstrate that the proposed approach attains a superior performance in comparison with other baselines, indicating that it is a more practical solution for autonomous racing. This work also suggests a transferable solution for other extreme-condition tasks. In the future, model confidence can be integrated to trigger a fallback strategy [30], thereby eliminating the requirement of manually designed physical constraints. Also, the proposed approach can be evaluated on broader racetracks to demonstrate its generalization ability.

References

[1] H. Liu, et al., "Hybrid-prediction integrated planning for autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 4, pp. 2597–2614, 2025.

[2] H. Yang, et al., "Human-guided continual learning for personalized decision-making of autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 5435-5447, 2025.

[3] C. C. Chung et al., "Autonomous System for Head-to-Head Race: Design, Implementation, and Analysis; Team KAIST at the Indy Autonomous Challenge," *IEEE Trans. Field Robot.*, vol. 2, pp. 574–600, 2024.

[4] A. Raji et al., "er.autopilot 1.1: A software stack for autonomous racing on oval and road course tracks," *IEEE Trans. Field Robot.*, vol. 1, pp. 332–359, 2024.

[5] J. Betz et al., "Human-like interactive behavior generation for autonomous vehicles: A Bayesian game-theoretic approach with turing test," *J. Field Robot.*, vol. 40, no. 4, pp. 783–809, 2023.

[6] J. Ni and J. Hu, "Dynamics control of autonomous vehicle at driving limits and experiment on an autonomous formula racing car," *Mech. Syst. Signal Process.*, vol. 90, pp. 154–174, 2017.

[7] H. Liu, et al., "Reinforced refinement with self-aware expansion for end-to-end autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, 2026.

[8] R. Zhao, et al., "A survey on recent advancements in autonomous driving using deep reinforcement learning: Applications, challenges, and solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 12, pp. 19365–19398, 2024.

[9] H. M. Eraqi, M. N. Moustafa, and J. Honer, "Dynamic conditional imitation learning for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 22988–23001, 2022.

[10] H. Liu, et al., "Augmenting reinforcement learning with Transformer-based scene representation learning for decision-making of autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 9, no. 3, pp. 4405–4421, 2024.

[11] J. Wu, et al., "Human-guided deep reinforcement learning for optimal decision making of autonomous vehicles," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 54, no. 11, pp. 6595–6609, 2024.

[12] J. Wu, et al., "Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14745–14759, 2023.

[13] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2011, pp. 627–635.

[14] M. Fazekas, Z. Demeter, J. Tóth, Á. Bogár-Németh, and G. Bári, "Evaluation of local planner-based stanley control in autonomous rc car racing series," in *Proc. IEEE Intell. Veh. Symp.*, 2024, pp. 252–257.

[15] A. Wischnewski, M. Euler, S. Gümüs, and B. Lohmann, "Tube model predictive control for an autonomous race car," *Veh. Syst. Dyn.*, vol. 60, no. 9, pp. 3151–3173, 2022.

[16] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3363–3370, 2019.

[17] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "A predictive safety filter for learning-based racing control," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7635–7642, 2021.

[18] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: A predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2713–2719, 2020.

[19] P. Cai, H. Wang, H. Huang, Y. Liu, and M. Liu, "Vision-based autonomous car racing using deep imitative reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7262–7269, 2021.

[20] B. D. Evans, H. A. Engelbrecht, and H. W. Jordaan, "High-speed autonomous racing using trajectory-aided deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 8, no. 9, pp. 5353–5359, 2023.

[21] E. Ghignone, N. Baumann, and M. Magno, "Tc-driver: A trajectory-conditioned reinforcement learning approach to zero-shot autonomous racing," *IEEE Trans. Field Robot.*, vol. 1, 527–536, 2024.

[22] A. Remonda, et al., "A simulation benchmark for autonomous racing with large-scale human data," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 102078–102100, 2025.

[23] J. Betz et al., "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 458-488, 2022.

[24] C. R. Qi, L. Yi, H. Su, and L. J., Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5105–5114, 2017.

[25] R. Rajamani, "Vehicle dynamics and control," 2nd ed., Boston, MA: Springer US, 2006.

[26] H. B. Pacejka, and E. Bakker, "The magic formula tyre model," *Veh. Syst. Dyn.*, vol. 21, no. S1, pp. 1–18, 1992.

[27] Y. Zhou, S. Wu, and C. Lv, "Multi-modal sensor fusion for localization for autonomous racing: A robust high-frequency estimation framework," in *Proc. Int. Conf. Adv. Robot. Mechatron.*, 2025, pp. 906–911.

[28] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit Q-learning," in *Int. Conf. Learn. Represent.*, 2022, pp. 1–11.

[29] S. Fujimoto, and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 20132–20145, 2021.

[30] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F. Y. Wang, "From Features Engineering to Scenarios Engineering for Trustworthy AI: I&I, C&C, and V&V," *IEEE Intell. Syst.*, vol. 37, no. 4, pp. 18–26, 2022.