IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

Vision-Based Policy Learning for High-Speed Autonomous Racing

Haoran Xu, *Student Member, IEEE*, Xianwei Chen, Yilin Lang, *Student Member, IEEE*, and Qinyuan Ren, *Senior Member, IEEE*

***Abstract*—Motion planning for autonomous vision-based car racing is a challenging task in robotics. Classical racing systems divide the task into numerous submodules, undermining computational efficiency and leading to error propagation. Previous studies have demonstrated impressive reinforcement learning (RL) results for end-to-end autonomous driving. However, RL exhibits poor scalability on high-dimensional data, such as images, and it is challenging to learn optimal racing behaviors due to a lack of global information about the environments. To address these issues, a two-phase learning paradigm is proposed in this work to train a vision-based racing policy. First, RL trains a teacher policy that integrates progress maximization with collision avoidance in the reward function and utilizes privileged information about the racetrack to achieve high-performance racing. Then, a student policy, relying only on an ego-centric depth camera for perception, is trained by distilling racing knowledge from the teacher policy. The student policy achieves high-speed drive, high success rate, and smooth control in vision-based racing games. The proposed approach is validated in the simulation and on a real-world 1/10-scale race car, showing that the approach outperforms previous model-based and learning-based baselines.**

***Index Terms*—Machine Learning for Robot Control, Motion Control, Reinforcement Learning**

I. Introduction

AUTONOMOUS racing has attracted much attention in robotics, serving as a testing ground for planning and control technologies of vehicles that move at the dynamic limits of handling [1]. The inherent characteristics of racing games provide challenges for autonomous systems due to nonlinear dynamics, noisy sensor data, and real-time computation requirements [2]. The racing task has two easy-to-measure evaluation metrics: lap time and success rate. Hence, the trade-off between high-speed driving behaviors and ensuring safety acts as a critical issue in planning and control algorithms [3]. Classic autonomous racing systems, which are model-based, generally decompose the racing task into trajectory planning and motion control [4]. However, these classical systems are always sensitive to unknown disturbances, inefficient in computation, and subject to error propagation [5].

End-to-end methods are widely used to solve the difficulties of classical systems in autonomous driving. Using deep neural networks, data-driven methods directly process raw sensing data and further output control commands. Previous studies based on reinforcement learning (RL) exhibit exceptional performance in end-to-end autonomous racing [6], [7]. In [8], a super-human racing policy in Gran Turismo Sport is learned with the Soft Actor-Critic (SAC) algorithm, which incorporates prior knowledge of the racetrack as input. [9] proposes a trajectory-aided learning (TAL) approach for high-performance racing, integrating RL with model-based motion control. In these approaches, additional racetrack information or guidance from model-based controllers accelerates the training process of RL and improves racing performance. Nevertheless, in racing games, human drivers depend only on local ego-centric observations to perceive the environment, such as visual information and proprioceptive estimation, which means global information about the racetrack is typically inaccessible in real-world applications. Therefore, an autonomous racing agent achieving impressive performances with local observations still warrants further exploration.

Fig. 1. The 1/10-scale race car prototype adopted in this study. The vehicle utilizes an ego-centric depth camera for perception of the environment. The policy processes depth images and proprioceptive states, then outputs commands to the electronic speed controller for motion control of the vehicle.

Numerous studies have been conducted on vision-based autonomous driving. A vision-based RL method is developed in [10] for autonomous racing agents. Nonetheless, the policy proposed in the study is unable to generate optimal racing trajectories consistently, and unsafe collisions may occur during driving. A hybrid approach combining imitation learning with RL proposed in [11] improves the success rate in real-

*This work was supported in part by the State Key Laboratory of Industrial Control Technology under Grant ICT2025A17 and in part by the Open Foundation of the National Key Laboratory of Autonomous Intelligent Unmanned Systems under Grant 2024-SRIAS-KF-007. *(Corresponding author: Qinyuan Ren.)*

All authors are with College of Control Science and Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China.

E-mail: renqinyuan@zju.edu.cn

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

world racing games, while the agent executes lower throttle in the real world than in the open-loop planning, resulting in suboptimality. Therefore, the absence of global information in the training process is identified as an impediment to learning optimal racing behaviors with only visual observations. [12], [13] decompose the vision-based high-speed control into a visual feature extraction module and a path-following controller, achieving good performance. Their controller is constrained by using 2D Lidar data, which could be unavailable in vision-based racing games. Additionally, [14] introduces an asymmetric actor-critic architecture to achieve super-human racing performance with visual input. However, the robustness of their racing system has not been validated in challenging, noisy real-world experiments. In [15], a privileged learning algorithm for perception-aware agile flight is introduced and validated in hardware-in-the-loop simulation. Due to being trained in fixed environments, the policy lacks generalizability to environments beyond the training settings, and the neural network cannot resolve the memory of historical observations. Hence, the sim-to-real gap contributes as another bottleneck to the practical applications of vision-based racing policies.

This paper proposes a two-phase learning method to address the challenge of learning optimal behaviors with local observations in real-world autonomous racing games. In the first phase, a teacher policy with access to privileged racetrack information is trained via RL. The reward function in RL combines progress maximization with collision avoidance against track boundaries, enabling the generation of optimal racing trajectories on the specified track. In the second phase, a student policy without global information is trained by distilling optimal driving behaviors from the teacher policy, avoiding abundant explorations in the high-dimensional space.

Different from previous work on distillation from privileged information [15], this paper primarily addresses the sim-to-real gap of depth images, which includes depth range, noisy depth values, and invalid depth artifacts. To this end, the depth images in the simulation are preprocessed to simulate depth noise, and a variational autoencoder (VAE) is pre-trained to reconstruct ground truth from noisy images, which facilitates the knowledge distillation and enhances the robustness of the policy. A recurrent neural network (RNN) is incorporated into the student policy to retain information from past observations and address partial observability. The resulting policy demonstrates better generalization performance with local noisy observations, including ego-centric depth images and proprioceptive estimation data. The real-world application is verified on a 1/10-scale race car prototype, as shown in Fig. 1. Experiments in simulation and real-world scenarios indicate that the proposed method can achieve high-speed drive, high success rates, and smooth control, outperforming previous model-based and learning-based baselines.

The main contributions of this paper are summarized as follows:

- The privileged information and a general racing reward are incorporated into the teacher policy, which promotes the generation of optimal racing trajectories.
- A racing knowledge distillation method is proposed, overcoming the difficulty of learning optimal racing behaviors with high-dimensional observations.
- A VAE-RNN network architecture is designed, which figures out the process of noisy images and the memory of local observations. The resulting policy achieves zero-shot sim-to-real transfer.

The rest of this paper is organized as follows. Section II introduces the problem statement of learning a vision-based racing policy. Section III proposes the two-phase learning scheme. Section IV represents the details of domain randomization in the simulation. Section V analyzes the experimental results. Finally, Section VI summarizes this paper and discusses future work.

II. Problem Statement

In vision-based autonomous racing, the agent partially perceives the environment with a front-facing camera. The racing task is formulated as a Partially Observable Markov Decision Process (POMDP) problem. The POMDP can be presented as a tuple $(\mathcal{S}, \mathcal{A}, \Omega, \mathcal{O}, \mathcal{T}, \mathcal{R})$, where $\boldsymbol{s} \in \mathcal{S}$ is the possible state of the racing agent and $\boldsymbol{u} \in \mathcal{A}$ is the possible action that includes the desired acceleration and steering; $\boldsymbol{o}^\dagger \in \Omega$ is the observation that includes the depth image and the proprioceptive state of the vehicle; $\mathcal{O} : \mathcal{S} \times \mathcal{A} \times \Omega \to [0, 1]$ represents the observation probability of a given state; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the transition probability of the system and the transition function is $\boldsymbol{s}_{t+1} \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{u}_t)$; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function defined to minimize the lap time and the collision loss. The action at each time step is generated by a control policy $\boldsymbol{u}_t \sim \pi^\dagger(\cdot|\boldsymbol{o}_t^\dagger)$ conditioned on the observation. The goal of the algorithm is to find a policy that minimizes the expected cumulative return $\mathbb{E}_{\boldsymbol{u}_t \sim \pi^\dagger(\cdot|\boldsymbol{o}_t^\dagger)} \left[\sum_{t=0}^{T-1} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{u}_t)\right]$, where γ is the discount factor, T is the episode length.

As discussed in Section I, RL is plagued with poor scalability on high-dimensional data, and it is challenging to learn optimal racing behaviors from local observations. To address this issue, a two-stage policy learning scheme is introduced: Firstly, a teacher policy $\pi^*(\boldsymbol{o}^*)$ with state-based privileged information $\boldsymbol{o}^*$ is trained using model-free RL:

$$\pi^* = \arg\max_{\pi^*} \mathbb{E}_{\boldsymbol{u}_t \sim \pi^*(\cdot|\boldsymbol{o}_t^*)} \left[\sum_{t=0}^{T-1} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{u}_t)\right]. \tag{1}$$

Then, a vision-based non-privileged student policy $\pi^\dagger(\boldsymbol{o}^\dagger)$ is distilled from the teacher policy:

$$\pi^\dagger = \arg\max_{\pi^\dagger} \mathcal{L}_{KD}(\pi^*, \pi^\dagger), \tag{2}$$

where $\mathcal{L}_{KD}$ is the loss function for knowledge distillation (KD), which measures the difference between the student policy and the teacher policy.

III. Methodology

This paper uses a two-phase training scheme to train adaptive end-to-end control policies, as shown in Fig. 2. The teacher policy has access to privileged information about the environment, i.e., 2D Lidar information and sampled points of the racetrack's centerline. This global information promotes

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

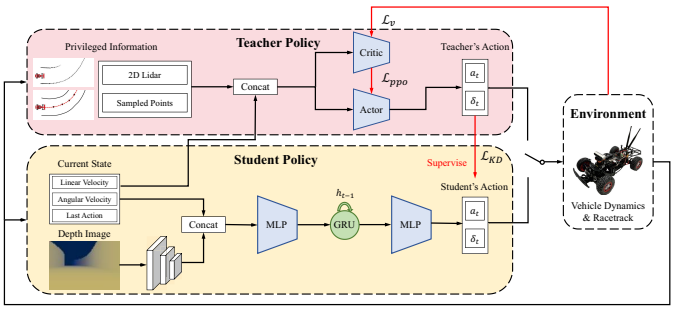


Fig. 2. System Overview. First, a teacher policy with privileged information is trained using model-free reinforcement learning. Then, the teacher policy is distilled into a vision-based student policy, which employs a convolutional neural network encoder for image processing and a recurrent neural network with the recurrent state h_{t-1} to retain the memory of historical observations. The privileged information is only accessible in the first learning stage. The agent relies only on the estimated state and a depth image to control in the second stage. The action of control policies includes the desired acceleration a_t and steering δ_t. $\mathcal{L}_v$ is the critic loss, $\mathcal{L}_{ppo}$ is the actor update objective, $\mathcal{L}_{KD}$ is the loss function of the knowledge distillation.

the generation of optimal racing trajectories. Then, a vision-based student policy, which only has access to depth images and proprioceptive states, is distilled from the teacher policy to learn optimal driving behaviors with local observations.

A. Racing Reward Function

The first training phase aims to train a teacher policy, which minimizes the lap time on a given racetrack. However, the race car has to finish driving a lap before receiving the time signal. To avoid instability of RL due to sparse reward, an approximate reward function based on the track progress is designed, which can be evaluated at each time step.

A sequence of n points $(\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n)$ is sampled along the racetrack's centerline. To calculate the track progress s_t of the race car's position $\boldsymbol{p}_t$ at each time step t, it is needful to find the nearest point $\boldsymbol{g}_t$ on the track between two adjacent points $\boldsymbol{q}_l$ and $\boldsymbol{q}_{l+1}$, as shown in Fig. 3. The optimization problem is formulated as:

$$\begin{aligned} \boldsymbol{g}_t, \boldsymbol{q}_l &= \arg\min_{\boldsymbol{g}_t, \boldsymbol{q}_l} \|\boldsymbol{g}_t - \boldsymbol{p}\|_2 \\ \text{s.t. } \boldsymbol{g}_t &= \boldsymbol{q}_l + \alpha(\boldsymbol{q}_{l+1} - \boldsymbol{q}_l), \\ \alpha &= \frac{(\boldsymbol{p} - \boldsymbol{q}_l) \cdot (\boldsymbol{q}_{l+1} - \boldsymbol{q}_l)}{\|\boldsymbol{q}_{l+1} - \boldsymbol{q}_l\|_2^2}, \\ l &\in \{1, \cdots, n-1\}, \alpha \in (0, 1]. \end{aligned} \tag{3}$$

The track progress s_t at time t is defined as:

$$s_t = s(\boldsymbol{p}_t) = \sum_{i=1}^{l-1} \|\boldsymbol{q}_{i+1} - \boldsymbol{q}_i\| + \|\boldsymbol{g}_t - \boldsymbol{q}_l\|. \tag{4}$$

Then, the reward $r(t)$ at time t is calculated as:

$$\begin{aligned} r(t) &= r_{track} + r_{crash} + r_\delta, \\ r_{track} &= s_t - s_{t-1}, \\ r_{crash} &= \begin{cases} -\lambda_c \|\boldsymbol{v}_t\|^2 & \text{(when collision)} \\ 0 & \text{(otherwise)} \end{cases}, \\ r_\delta &= -\lambda_\delta \|\delta_t - \delta_{t-1}\|. \end{aligned} \tag{5}$$

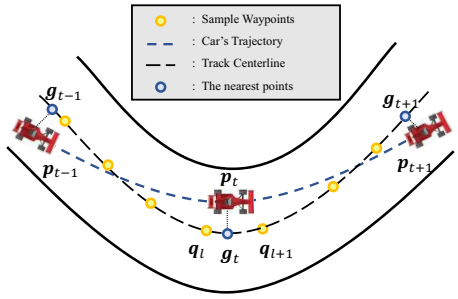


Fig. 3. The track progress s_t is calculated by finding the nearest point on the centerline of the track, equivalent to projecting the race car's position $\boldsymbol{p}_t$ on the line connecting two adjacent sampling points $\boldsymbol{q}_l$ and $\boldsymbol{q}_{l+1}$.

The penalty term r_{crash} guarantees that the vehicle only moves in the region of the track, where $\boldsymbol{v}_t \in \mathbb{R}^2$ is the linear velocity of the vehicle. This kinetic energy-based reward is introduced in [8]. The last term r_δ contributes to avoiding shaky steering command output, which is harmful to real-world applications. λ_δ, λ_c are positive parameters weighting the last two terms.

B. Learning Teacher Policy with Privileged Information

The privileged information is introduced in the teacher policy to guide the agent's behavior. The observation $\boldsymbol{o}_t^*$ at a given time step t is selected as $\boldsymbol{o}_t^* = [\boldsymbol{v}_t, \dot{\theta}_t, \boldsymbol{u}_{t-1}, \boldsymbol{d}_t, \boldsymbol{w}_t]$, where $\boldsymbol{v}_t \in \mathbb{R}^2$ is the linear velocity in the vehicle's body frame; $\dot{\theta}_t \in \mathbb{R}$ is the angular velocity of the vehicle's heading angle; $\boldsymbol{u}_{t-1}$ is the previous control command. Distance measurements $\boldsymbol{d}_t \in \mathbb{R}^{N_l}$ are acquired from a 2D Lidar that senses objects in front of the vehicle at a maximum distance d_{max} within the field of view (FOV) θ_{fov}. Point vector $\boldsymbol{w}_t \in \mathbb{R}^{2N_w}$ consists of N_w preceding sampled points of the centerline, and the distance between any two adjacent points is Δd. A z-score normalization is applied to all futures in the agent's observation. The output of the policy network $\boldsymbol{u}_t$ consists of the desired acceleration a_t and steering δ_t, all normalized to $[-1, 1]$. The maximum acceleration is a_{max}, and the maximum steering angle is δ_{max}.

The privileged information in the simulation refers to the future sampled points $\boldsymbol{w}_t$ and the distance measurements $\boldsymbol{d}_t$, which provide the agent with global information about the racing environment. Therefore, the environment is fully observable, and the POMDP transfers to a Markov Decision Process (MDP) so that the teacher policy does not need to memorize historical observations.

The teacher policy is updated by maximizing the PPO clip-objective [16]:

$$\mathcal{L}_{ppo} = \mathbb{E}_{\pi^*_{old}} \left[\min(\rho_{\pi^*} \hat{A}_t, clip(\rho_{\pi^*}, 1-\epsilon, 1+\epsilon)\hat{A}_t) \right], \tag{6}$$

where $\rho_{\pi^*} = \dfrac{\pi^*(\boldsymbol{o}_t^*)}{\pi^*_{old}(\boldsymbol{o}_t^*)}$, $\epsilon = 0.2$ is a hyperparameter, and $\hat{A}_t$ is calculated by Generalized Advantage Estimation (GAE). The critic loss for the value function $V^*(\boldsymbol{o}_t^*)$ is defined as:

$$\mathcal{L}_v = \mathbb{E}_{\pi^*_{old}} \left[\frac{1}{2} (r_t + \gamma V^*(\boldsymbol{o}_{t+1}^*) - V^*(\boldsymbol{o}_t^*))^2 \right]. \tag{7}$$

The actor and critic networks are all updated using the trajectories collected by the last teacher policy π^*_{old}. The training

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

process is efficient because the teacher policy does not process image information.

Remark 1: The optimality of racing trajectories refers to the policy's ability to maximize the expected cumulative reward as defined in the RL framework. This reward function is specifically designed to ensure that the racing trajectories generated are both high-performing and safe. The optimality of the teacher policy is achieved by utilizing privileged information in RL, which provides the policy with comprehensive environmental knowledge and enables the policy to make decisions in fully observable environments. This clarifies that the optimal solution is relative to the defined reward function and the availability of prior information, rather than an absolute global optimum.

C. Network Architecture of Student Policy

The student policy cannot access privileged information, so the track's centerline and boundary must be inferred solely from the visual information. The observation of the student policy is $\boldsymbol{o}_t^\dagger = [\boldsymbol{I}_t, \boldsymbol{\gamma}_t]$, where $\boldsymbol{I}_t$ is the depth image and $\boldsymbol{\gamma}_t = \left[\boldsymbol{v}_t, \dot{\theta}_t, \boldsymbol{u}_{t-1}\right]$ is the proprioceptive state of the vehicle that is part of $\boldsymbol{o}_t^*$. An RNN-based network architecture is designed to process local observations.

The Gated Recurrent Unit (GRU) [17] constitutes a recurrent state that efficiently captures dependencies in sequential data. The network architecture of the student policy $\pi^\dagger$ is shown in Fig. 2. The depth image $\boldsymbol{I}_t \in \mathbb{R}^{N_1 \times N_2}$ is processed with an image encoder q_ψ to obtain the latent vector $\boldsymbol{l}_t \in \mathbb{R}^{N_z}$ and then it is concatenated with the proprioceptive state $\boldsymbol{\gamma}_t$ to feed into a multilayer perceptron p_1. A low-dimensional embedding feature $\boldsymbol{\beta}_t \in \mathbb{R}^{N_\beta}$ is extracted by p_1 and then passed to a GRU component p_{GRU} with a recurrent state $\boldsymbol{h}_{t-1} \in \mathbb{R}^{N_h}$ to implicitly estimate the shape of the race track and the dynamic states of the vehicle. Finally, another multilayer perceptron p_2, similar to the teacher policy, outputs the control command $\boldsymbol{u}_t$ while receiving the recurrent state $\boldsymbol{h}_t$. The network architecture is written as follows:

$$\begin{aligned} &\text{Image Encoder:} && \boldsymbol{l}_t \sim q_\psi(\boldsymbol{I}_t), \\ &\text{Feature Embedding:} && \boldsymbol{\beta}_t = p_1(\boldsymbol{l}_t, \boldsymbol{\gamma}_t), \\ &\text{Observation Memory:} && \boldsymbol{h}_t = p_{GRU}(\boldsymbol{\beta}_t, \boldsymbol{h}_{t-1}), \\ &\text{Control Policy:} && \boldsymbol{u}_t = p_2(\boldsymbol{h}_t). \end{aligned} \tag{8}$$

To enhance the robustness of the student policy to noisy visual observations, the image encoder is pre-trained by a variational autoencoder (VAE) [18]. The encoder extracts the distribution of the latent representation $\boldsymbol{l}_t$, while the decoder reconstructs the depth image $\boldsymbol{I}_t^{rec}$. The VAE is trained using images collected by the teacher policy π^*. The loss function is detailed as follows:

$$\mathcal{L}_\psi = \mathbb{E}_{(\boldsymbol{I}_t, \tilde{\boldsymbol{I}}_t)\sim\mathcal{B}}\left[\|\tilde{\boldsymbol{I}}_t - \boldsymbol{I}_t^{rec}\|_2^2\right] + \lambda_{KL} D_{KL}(q_\psi(\cdot \mid \boldsymbol{I}_t)\|p(\boldsymbol{l}_t)), \tag{9}$$

where ψ denotes the parameter of the VAE, $\mathcal{B}$ is the depth image data buffer. The first term presents the reconstruction loss. The second term denotes the Kullback-Leibler divergence between the encoder $q_\psi(\cdot \mid \boldsymbol{I}_t)$ and a standard normal distribution $p(\boldsymbol{l}_t) = \mathcal{N}(0, I)$, with $\lambda_{KL} > 0$ is a weighting parameter. It is noteworthy that the noisy depth image $\boldsymbol{I}_t$ is fed into the encoder, and the original depth image $\tilde{\boldsymbol{I}}_t$ is used to supervise the reconstruction, as shown in Fig. 4(a). The difference between $\boldsymbol{I}_t$ and $\tilde{\boldsymbol{I}}_t$ will be explained in Section IV. This setting enables the encoder to denoise the images and achieve more accurate feature extraction.

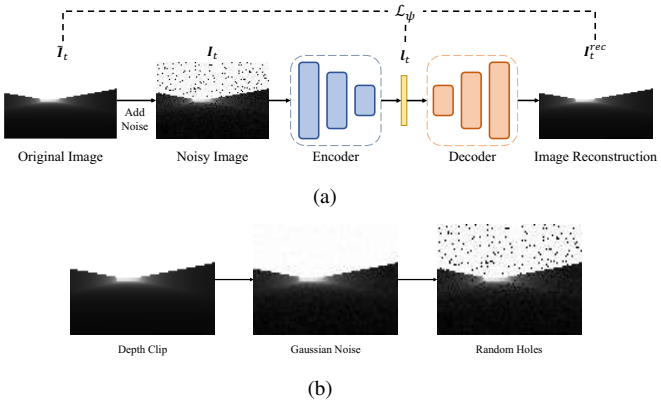


Fig. 4. (a) The architecture of feature extraction with denoising VAE. (b) The preprocessing of depth images in the simulation includes depth clip, Gaussian noise, and random holes.

Algorithm 1: Two-phase policy learning algorithm

Randomly initialize parameters of the teacher policy π^*, the student policy $\pi^\dagger$ and the VAE;
Initialize dataset $\mathcal{D}, \mathcal{B}$;
while *task not learned* **do**
  `// Reinforcement Learning`
  while *not converged* **do**
    Sample trajectories by teacher policy π^*;
    Update V^* via loss $\mathcal{L}_v$ in Eq. (7);
    Update π^* via objective $\mathcal{L}_{ppo}$ in Eq. (6);
  end
  `// VAE Training`
  Collect depth image dataset $\mathcal{B}$ by π^*;
  Train VAE on $\mathcal{B}$ via loss $\mathcal{L}_\psi$ in Eq. (9);
  `// Knowledge Distillation`
  while *not converged* **do**
    Sample trajectories tr by student policy $\pi^\dagger$;
    Collect actions given by the teacher policy $\mathcal{D}_i \leftarrow \left\{(\boldsymbol{o}_t^*, \boldsymbol{o}_t^\dagger, \pi^*(\boldsymbol{o}_t^*)) : (\boldsymbol{o}_t^*, \boldsymbol{o}_t^\dagger) \in tr\right\}$;
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$;
    Update $\pi^\dagger$ on $\mathcal{D}$ via loss $\mathcal{L}_{KD}$ in Eq. (10);
  end
end

D. Racing Knowledge Distillation

Since the teacher policy explicitly maps observations to actions, it can be used for action annotation of visual information. In the second training phase, a supervised learning method is utilized to learn optimal racing behaviors by minimizing the deviation between the teacher policy and the student policy using the objective:

$$\mathcal{L}_{KD} = \mathbb{E}_{(\boldsymbol{o}_t^*, \boldsymbol{o}_t^\dagger, \pi^*(\boldsymbol{o}_t^*))\sim\mathcal{D}}\left[\|\pi^*(\boldsymbol{o}_t^*) - \pi^\dagger(\boldsymbol{o}_t^\dagger)\|_2^2\right], \tag{10}$$

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

TABLE I
DOMAIN RANDOMIZATION OF PARAMETERS IN THE SIMULATION.

	Variable	Unit	Value Randomization
Dynamics	Mass of race car	$[kg]$	$\mathcal{U}(3.90, 3.95)$
	Inertia of race car	$[kg \cdot m^2]$	$\mathcal{U}(0.046, 0.048)$
	Friction coefficient	$[-]$	$\mathcal{U}(0.7, 0.9)$
	Cornering stiffness of front wheel	$[-]$	$\mathcal{U}(4.5, 4.7)$
	Cornering stiffness of rear wheel	$[-]$	$\mathcal{U}(5.3, 5.5)$
	Actuator delay	$[s]$	$\mathcal{U}(0.005, 0.01)$
	Acceleration error	$[m/s^2]$	$\mathcal{N}(0.0, 0.1^2)$
	Steering error	$[rad]$	$\mathcal{N}(0.0, 0.02^2)$
	Variable	**Unit**	**Noise Randomization**
Perception	Lidar scan	$[m]$	$\mathcal{N}(0.0, 0.01^2)$
	Depth image	$[m]$	$\mathcal{N}(0.0, 0.04^2)$
	Linear Velocity	$[m/s]$	$\mathcal{N}(0.0, 0.1^2)$
	Angular Velocity	$[rad/s]$	$\mathcal{N}(0.0, 0.2^2)$

where $\mathcal{D}$ is the observation-action data buffer, $\pi^*(\boldsymbol{o}_t^*)$ is the action of the teacher policy and $\pi^\dagger(\boldsymbol{o}_t^\dagger)$ is the action of the student policy. The dataset aggregation (DAgger) [19] is leveraged to train the student policy $\pi^\dagger$, rolling out the student policy for data collection and running the teacher policy offline for action ground truth. This supervised learning method avoids abundant explorations in the high-dimensional space and accelerates the training process. The whole training process of the proposed two-phase learning scheme is summarized in Alg. 1.

IV. SIMULATION AND DOMAIN RANDOMIZATION

Transferring the student policy directly into the real world is challenging because of the sim-to-real gap. In the vision-based high-speed racing task, the sim-to-real gap consists of two parts: the dynamics of the race car and the noisy sensory observations. To achieve zero-shot transfer, the domain randomization technique is employed in the simulation.

Vehicle Dynamics: First, the simulation uses the nominal vehicle dynamics [3] and forward integrates them using a 4th-order Runge-Kutta scheme. The parameters of the dynamic model are identified using real-world data collected by a precise motion capture system. When training policies, the key parameters are randomized to simulate complex real-world dynamics, as shown in Table I, where $\mathcal{N}(\cdot, \cdot)$ denotes a Gaussian distribution and $\mathcal{U}(\cdot, \cdot)$ denotes a uniform distribution. In the simulation, the maximum acceleration is $a_{max} = 8m/s^2$, the maximum steering angle is $\delta_{max} = 0.4rad$, and the maximum speed is $v_{max} = 8m/s$.

Privileged Information Simulation: The 2D Lidar scans received by the teacher policy are simulated by tracing several rays from the vehicle's position to different directions. Distance measurements, which have a maximum value of $d_{max} = 15m$ and an FOV of $\theta_{fov} = 270°$, are calculated when the rays collide with the racetrack. The number of rays is 1080, which is divided into $N_l = 72$ partitions, corresponding to an angular resolution of $3.75°$. Then, the minimum distance in each partition is used as the value of this partition, and all of these are combined as $\boldsymbol{d}_t$. The future points $\boldsymbol{w}_t$ are sampled from the pre-extracted centerline of the racetrack, having a total number of $N_w = 30$ and a resolution of $\Delta d = 0.2m$. Furthermore, noise is also added to the state estimation data in the simulation, as shown in Table I.

TABLE II
HYPERPARAMETERS OF EACH TRAINING STAGE, WHERE RL REFERS TO REINFORCEMENT LEARNING, VAE REFERS TO PRE-TRAINING OF VAE, AND KD REFERS TO KNOWLEDGE DISTILLATION.

	Hyperparameter	Value
RL	Weight of shaky steering penalty λ_δ	0.2
	Weight of collision penalty λ_c	0.3
	Clip ratio ϵ	0.2
	GAE lambda	0.95
	Learning rate	0.0003
	Discount factor γ	0.99
	Batch size	512
	Optimizer	Adam
VAE	Dimension of latent vector N_z	64
	Weight of KL divergence λ_{KL}	0.001
	Buffer size of $\mathcal{B}$	80000
	Batch size	128
	Learning rate	0.0003
	Optimizer	Adam
KD	Dimension of embedding feature N_β	64
	Dimension of hidden state N_h	256
	Buffer size of $\mathcal{D}$	1024
	Batch size N_{bs}	64
	Length of each trajectory $T_{student}$	128
	Learning rate	0.0003
	Optimizer	Adam

Depth Image Simulation: The depth images received by the student policy are acquired from the Pybullet simulator, which uses the position and orientation of the vehicle and a URDF model of the racetrack to update images. To bridge the visual sim-to-real gap, the images are preprocessed by applying depth clip, Gaussian noise, and random holes, as shown in Fig. 4(b). The clipped depth image is represented by $\tilde{\boldsymbol{I}}_t$, and the final noisy depth image is represented by $\boldsymbol{I}_t$. The depth images in the simulation and the real world both have a resolution of 96×64, an FOV of $87° \times 58°$, and a range of $0.28m - 5.0m$.

V. EXPERIMENTS

A. Experimental Setup

The proposed method is evaluated in the simulation and the real world. First, baselines are compared in the simulation environment. Second, the cross-validation and comparison with other RL methods demonstrate the generalizability and data scalability of the proposed method. Furthermore, ablation experiments are conducted to study the impact of key components. Finally, the student policy is deployed in real-world experiments to verify the algorithm's feasibility.

1) Training Details: The hyperparameters of each learning phase are detailed in Table II. In each training stage, when rolling out policies to collect trajectories, the race car is placed at a random position on the racetrack, and the frequency of all policies is 30Hz. The teacher policy is used to collect images

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

TABLE III
QUANTIFIABLE CHARACTERISTICS OF FOUR MAPS USED IN THE SIMULATION EXPERIMENTS.

Map	AUT	ESP	GBR	MCO
Track length [m]	94.90	236.93	201.84	178.71
Straight percentage [%]	64.92	58.98	59.45	60.60
Corner count [-]	7	7	7	16

for pre-training of the image encoder. The image buffer $\mathcal{B}$ consists of both noisy depth images and original clipped depth images. The knowledge distillation utilizes the student policy to collect the observation-action buffer $\mathcal{D}$. In each training epoch, we sample N_{bs} trajectories from $\mathcal{D}$ to calculate the loss function $\mathcal{L}_{KD}$, each having $T_{student}$ time steps. The hidden state $\boldsymbol{h}_t$ propagates through time steps. The simulation and the training of policies are conducted on a workstation with an i9-10980XE CPU and an NVIDIA RTX 3090 GPU.

2) Simulation Environment: The proposed policies are trained using the open-source F1TENTH simulator [6], which is modeled on the Gym style environment and the internal dynamics updates at 100 Hz. Figure 5(a) shows the shapes of the four maps used in the experiments: AUT, ESP, GBR, and MCO. Quantifiable characteristics of four maps are measured and summarized in Table III, including track length, straight percentage, and the number of corners.

3) Real World: As shown in Fig. 1, the hardware platform consists of an off-the-shelf model race car chassis with a Traxxas Velineon 3351R brushless DC electric motor, which is driven by a VESC 6MkIV electronic speed controller (ESC). The depth images are captured by an Intel RealSense D435 camera. Onboard computation and control tasks are performed on an NVIDIA Jetson TX2. The desired accelerations are processed by integration to get the desired speed values.

4) Evaluation metrics: Three metrics are chosen to evaluate each method:

- Lap time, which shows the racing performance;
- Success rate, defined as the ratio of finishing the whole track in all runs, indicating the ability of collision avoidance;
- Mean jerk of the racing trajectory, defined as the time derivative of acceleration, indicating the smoothness of control output.

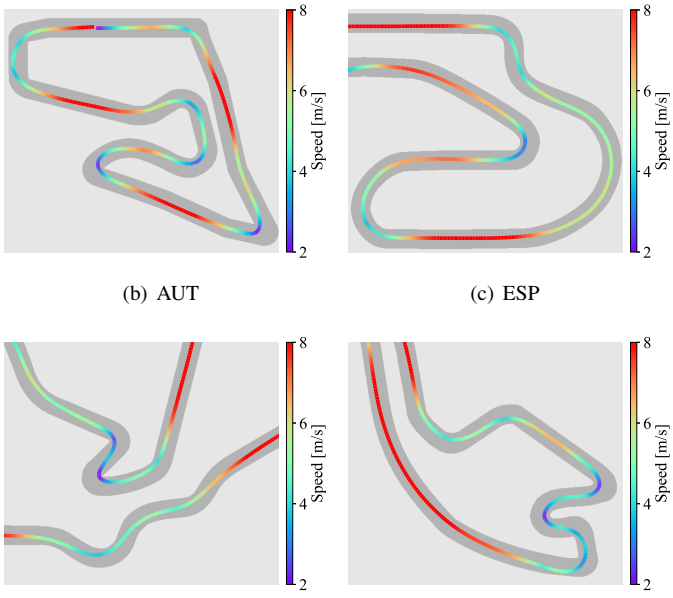
B. Baseline Comparisons

This section demonstrates that the vision-based student policy can reliably drive the vehicle on complex racetracks without privileged information about the racetrack. Example trajectories produced by the student policy in different racetracks are shown in Fig. 5. The baselines are selected as[1]:

- Optimization-based method with prior knowledge about racetrack and vehicle dynamics (**MPCC**) [3], with the input of global state estimation of the vehicle and parameterized centerline and boundary of the racetrack;

[1]MPCC and TAL are based on the open-source repository: https://github.com/BDEvan5/f1tenth_benchmarks.

(a) Maps of the AUT, ESP, GBR, and MCO (left to right) racetracks.



(b) AUT (c) ESP

(d) GBR (e) MCO

Fig. 5. Example trajectories produced by the proposed vision-based policy in the selected environments used for evaluation.

TABLE IV
MEAN LAP TIMES[S] AND STANDARD DEVIATIONS FOR DIFFERENT METHODS TESTED ON FOUR MAPS. THE BOLD FONT HIGHLIGHTS THE BEST VALUE IN EACH COLUMN.

Method	**Map**			
	AUT	ESP	GBR	MCO
MPCC	16.93±0.42	39.57±0.36	35.42±0.57	32.07±0.49
TAL	19.67±0.54	45.84±0.49	39.67±0.78	35.04±0.89
Teacher	**16.19±0.46**	37.84±0.38	**32.57±0.47**	**29.72±0.41**
Student	16.27±0.47	**37.74±0.41**	32.84±0.44	30.07±0.43

- Data-driven approach using trajectory-aided learning (**TAL**) [9], with the input of Lidar scans from current and previous time steps;
- The proposed teacher policy trained by RL (**Teacher**), with the input of privileged information and proprioceptive state;
- The proposed student policy trained by knowledge distillation (**Student**), with the input of depth images and proprioceptive state.

Four different racetracks (AUT, ESP, GBR, MCO) are used for evaluation, and all methods are tested 40 times with different starting points.

1) Lap time: Table IV presents the lap times of the four racing methods evaluated on each track. Among the baselines, the TAL method exhibits the longest lap times. The lack of racing encouragement in the reward function and the imitation error with the model-based method contribute to the TAL agent's suboptimal behaviors. In contrast, the MPCC method

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

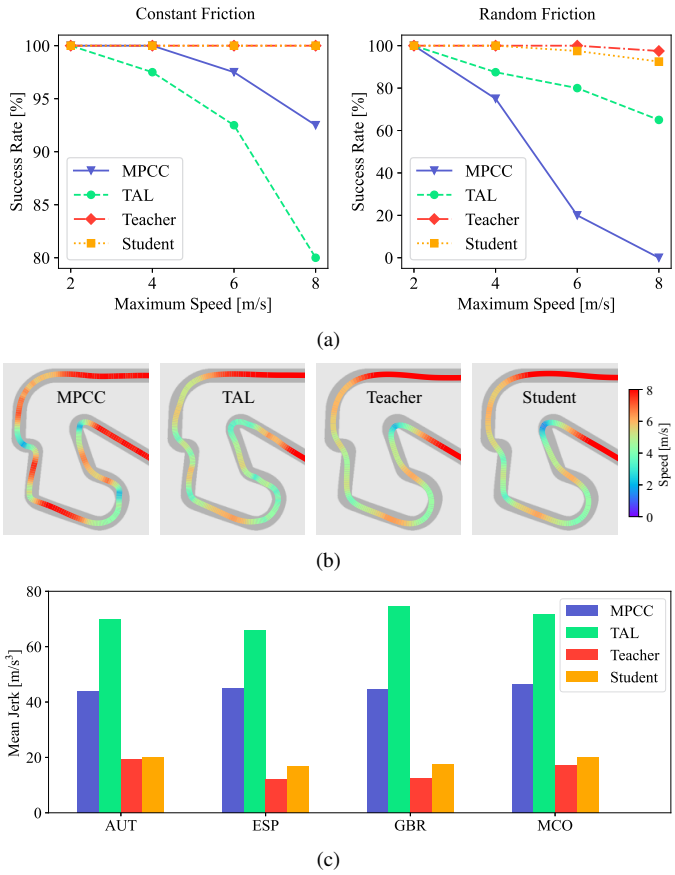


Fig. 6. (a) Success rates of different methods with different maximum speeds tested on the MCO track. The left image results from settings in constant friction, and the right image results from settings in random friction. (b) Trajectories with speed information of four methods tested on the ESP track. The proposed policies achieve smoother speed control commands. (c) Mean jerk of racing trajectories produced by different methods tested on all maps.

benefits from prior knowledge of the entire racetrack, resulting in improved racing performance. However, it struggles to balance between maximizing racing progress and minimizing contouring error, leading to relatively conservative trajectories. The teacher policy with privileged information achieves the shortest lap times due to reward shaping and guidance from future centerline waypoints. Without prior knowledge about the race track, the vision-based student policy obtains similar lap times compared to those of the teacher policy, outperforming the model-based control and the data-driven baseline.

2) Success rate: To evaluate the baselines' ability of collision avoidance, each method is tested with different maximum speeds on the MCO track. The experiment is divided into two groups, one with a constant tire-surface friction $\mu = 0.8$ and the other with a random friction $\mathcal{N}(\mu, \sigma^2)$, where $\sigma = 0.1$, so that the robustness to the tire-surface friction can be seen. The success rates are shown in Fig. 6(a). When encountering constant friction, all methods demonstrate a high success rate over 80%, and the proposed policies perform best. However, the success rate of the MPCC controller is significantly diminished under random friction. Due to the high reliance on an accurate dynamic model, such a classical control system is sensitive to unknown disturbances. As a model-free data-driven approach, the TAL method performs better. However, since it only imitates a model-based controller to track a preset trajectory, the TAL fails to achieve consistently high success rates. The teacher policy attains the highest success rate under random friction, benefiting from the collision penalty and privileged information about the upcoming track boundaries. The proposed vision-based student policy, despite relying on a limited range of perception, successfully completes the racing task. Even under random friction, it achieves a 92.5% success rate with a maximum speed of 8m/s.

3) Mean jerk: The smoothness of the trajectory is an essential metric for evaluating racing performance, significant in real-world applications. We test all methods in four maps for a lap and record the physical dynamics data of the racing trajectories. Figure 6(b) shows trajectories with speed information of four methods tested on the ESP track, and Fig. 6(c) shows the mean jerk of all trajectories on all maps. Experimental results show that the TAL method often produces non-smooth trajectories, mainly due to the absence of control smoothness terms in its reward function. The racing trajectories of the MPCC controller are slightly smoother, as its optimization problem maximizes the racing progress under the constraint of the control commands. The teacher policy achieves the smoothest racing trajectories, showing the effectiveness of the penalty reward on shaky steering commands. When entering curves, the teacher policy can break or adjust the steering in advance, as RL leverages the shape of the racetrack in the observation to drive smoothly while reducing the lap time. Moreover, the vision-based student policy also drives a smooth trajectory, achieving a mean jerk below $25m/s^3$ on all maps, which is beneficial for real-world applications.

C. Cross-Validation and Data Scalability

To demonstrate the generalization performance of our method, a cross-validation experiment is also conducted. The teacher and student policies are trained on a single map and tested on all four maps. The generalization results are shown in Table V. From the results, we can see that all policies have the ability to finish a lap on unseen racetracks. When tested on the training map, the policies always achieve the shortest lap times, the smoothest control, and the highest success rates. When encountering unseen racetracks, the performance of the policies inevitably declines, especially for those trained on simple racetracks, such as the AUT map. The policy trained on the AUT map sometimes collides with the racetrack, causing a low success rate. However, due to the challenging corner at the bottom right of the AUT map, the policy trained on the AUT still outperforms other policies when tested on the AUT map. Comparing policies trained on all four maps, the policy trained on the MCO map shows the best generalization ability. It can always drive the vehicle at high speed and avoid unsafe behaviors, which can be attributed to the complexity of the MCO map and its numerous corners.

To demonstrate the data scalability of our method on high-dimensional observation spaces, two vision-based RL methods are compared with the proposed method on the same racing task: (1) a CNN-based policy network trained by the PPO algorithm, which integrates an RNN to solve the partially

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

TABLE V
THE CROSS-VALIDATION RESULTS OF VISION-BASED STUDENT POLICIES TRAINED ON A SINGLE MAP AND TESTED ON DIFFERENT MAPS, THE EVALUATION METRICS ARE LAP TIME (L.T.), MEAN JERK (M.J.), AND SUCCESS RATE (S.R.).

Train Map	Test Map AUT			Test Map ESP			Test Map GBR			Test Map MCO		
	L.T. $[s]$	M.J. $[m/s^3]$	S.R. [%]	L.T. $[s]$	M.J. $[m/s^3]$	S.R. [%]	L.T. $[s]$	M.J. $[m/s^3]$	S.R. [%]	L.T. $[s]$	M.J. $[m/s^3]$	S.R. [%]
AUT	16.08±0.36	19.89±0.68	100	38.81±0.62	18.35±0.87	80.0	34.34±0.64	19.43±0.83	52.5	31.57±0.67	22.79±0.76	37.5
ESP	16.84±0.55	24.04±1.12	60.0	37.18±0.32	16.83±0.53	100	33.65±0.53	18.27±0.64	70.0	30.64±0.47	21.64±0.58	62.5
GBR	16.42±0.44	22.25±0.82	67.5	37.79±0.42	16.94±0.62	90.0	32.47±0.32	17.52±0.57	100	29.58±0.28	20.56±0.62	80.0
MCO	16.24±0.41	21.87±0.70	77.5	37.81±0.49	17.15±0.57	92.5	32.64±0.36	17.68±0.65	95.0	29.35±0.24	19.87±0.53	100

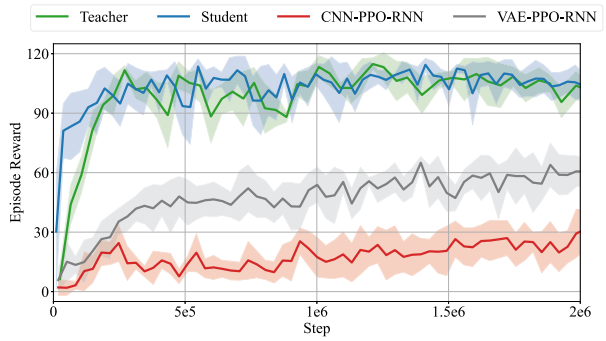


Fig. 7. The comparison of episode reward in the training process of the proposed policies and vision-based RL methods, the shadow refers to the standard deviation.

observable decision problem (CNN-PPO-RNN); (2) a policy network receiving features extracted by a pre-trained VAE and trained by the PPO algorithm, which also contains an RNN (VAE-PPO-RNN). The returned episode rewards of all policies in the process of training are shown in Fig. 7. An episode is terminated when it reaches a maximum length of 20s or a collision occurs during the racing game.

From the results, we can see that vision-based RL methods struggle to train the policies stably and find the optimal solution for racing games. The CNN-PPO-RNN policy displays the most disappointing performance in the training process; the episode reward is unstable and does not converge within 2×10^6 time steps of interacting with the environment. The resulting policy is often unable to complete a lap and frequently collides with the racetrack. The VAE-PPO-RNN policy achieves a slightly better performance, attributable to its operation within a low-dimensional space. However, due to the partial observations of the environment and the lack of global guidance to drive the race car, the policy is still unable to drive optimally. Thanks to the guidance of privileged information, the teacher policy only needs to make decisions in MDP and receives state-based low-dimensional observations. Using only 8×10^5 time steps of interacting with the environment, the teacher policy is ready to drive the car to finish the lap without collision. The process of training is stable and efficient, and the resulting policy can outperform other baselines. The knowledge distillation is even more efficient; the student policy utilizes only 6×10^5 interacting time steps to converge and achieves a similar performance to the teacher policy in spite of the necessity of processing the partially observable and high-dimensional visual observations. Therefore, the pre-training of VAE enables the algorithm to operate in a low-dimensional space, reducing computational complexity; the knowledge distillation facilitates the algorithm's ability to learn optimal behaviors with historical hidden states, avoiding abundant explorations. These two components of our method are crucial for data scalability.

D. Ablation Study

An ablation study evaluates the impact of the GRU component, privileged information, reward shaping, VAE pertaining, and VAE architecture. These policies are all tested on the most challenging MCO map. The results of three evaluation metrics are presented in Table VI.

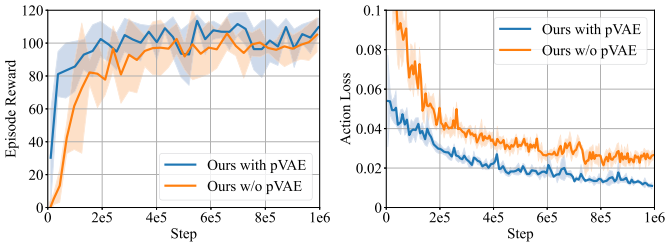
1) GRU component: According to the results, supervised by the same teacher policy, the proposed vision-based policy performs significantly better than the policy without the GRU component (w/o GRU). The policy without GRU exhibits larger action errors when distilled from the teacher policy, resulting in frequent cutting corners and the absence of advanced braking or steering, in contrast to the teacher policy. Due to the limited FOV of the camera, the policy sometimes cannot react in time and collides with the track boundary. It achieves a lower success rate (67.5%), a longer lap time (32.03s), and a higher mean jerk ($28.28 m/s^3$). It can be inferred that the GRU component could help the agent anticipate the shape of the racetrack and take the right driving actions in advance. The proposed architecture enables the policy to leverage past observations and contributes to robust and adaptive racing performance.

2) Privileged Information: Without some privileged information (w/o P.I.), e.g., the future sampled points $\boldsymbol{w}_t$, the teacher policy does not know the centerline in the future and drives the vehicle only using the distance measurement. From the results, although the teacher policy still achieves a high success rate, the lap time and the mean jerk indicate that its racing performance is deeply affected. When entering curves, the teacher policy is unable to anticipate the centerline's direction and often outputs a large steering change at the corner. This driving behavior is also distilled into the student policy, which results in a long lap time (33.19s) and a high mean jerk ($26.79 m/s^3$). Due to the action errors, the student policy drives a riskier trajectory than the teacher policy, leading to a lower success rate. It can be concluded that privileged information plays a significant role in the proposed method,

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

TABLE VI
ABLATION STUDY OF VARIOUS POLICIES ON SUCCESS RATE (S.R.), LAP TIME (L.T.), AND MEAN JERK (M.J.).

		S.R.[%]	L.T.[s]	M.J.[m/s^3]
Proposed Policy	Teacher	100	29.72	20.07
	Student	100	30.07	20.43
w/o GRU	Student	67.5	32.03	28.28
w/o P.I.	Teacher	100	32.57	23.43
	Student	85.0	33.19	26.79
CTH reward	Teacher	100	33.61	28.62
	Student	100	33.94	29.27
w/o pVAE	Student	90.0	30.57	22.89
VAE feature	$N_z = 32$	95.0	31.29	21.87
	$N_z = 64$	100	30.07	20.43
	$N_z = 128$	100	30.12	20.17



(a) Episode Reward (b) Action Loss

Fig. 8. The episode reward and action loss in the training process of the student policy with and without pre-trained VAE (pVAE), the shadow refers to the standard deviation.

which helps with the perception of the racetrack and guides the agent to drive a faster and safer racing trajectory.

3) Reward Shaping: To illustrate the influence of reward shaping, we replace our reward function in the teacher policy with a cross-track and heading error (CTH) reward and retrain the teacher and student policy. The CTH reward is formulated as $r_{CTH} = \frac{\|\boldsymbol{v}_t\|}{v_{max}} \cdot \cos\phi_c - d_c + r_{crash}$, where $\boldsymbol{v}_t$ is the velocity of the vehicle, $v_{max} = 8m/s$ is the maximum speed, ϕ_c is the heading error with the centerline, d_c is the cross-track error, r_{crash} is defined in Eq. (5). As the result shows, due to the penalty on cross-track error in the CTH reward, the teacher policy performs a longer lap time (33.61s) and a much higher mean jerk ($28.62m/s^3$), similar to a path-following controller. The distilled student policy also performs similarly; however, it is noticed that the student policy from the CTH reward seldom collides with the racetrack, because the cross-track penalty constrains the vehicle from being far from the centerline, resulting in a more conservative controller.

4) VAE Pre-training and VAE Architecture: To display the robustness of the student policy with pre-trained VAE, we compare the proposed student policy with a CNN-based policy trained purely by imitation loss in Eq. (10) (w/o pVAE). Two policies have the same encoder architecture and utilize the same teacher policy to distill racing behaviors. The training process of the two policies is shown in Fig. 8. Due to the pre-trained image encoder, the proposed student policy has an advantage in data efficiency in distilling racing knowledge. Two policies achieve similar episode rewards after $6e^5$ steps of training because they can all access the optimal control policy. However, since the pre-trained VAE has the ability to process noisy images, the proposed student policy exhibits a significantly lower action loss, as shown in Fig. 8(b). Furthermore, when tested on the same MCO map, the policy without pre-trained VAE demonstrates a lower success rate and a bigger mean jerk. It can be concluded that the pre-trained VAE can extract a more accurate feature, facilitating the process of knowledge distillation and enhancing the robustness of the student policy, which is further verified in Section V-E.

Additionally, we explore the impact of the dimension of the VAE feature to validate the design choice. Here, we compare the default setting ($N_z = 64$) with two feature sizes: $N_z = 32$ and $N_z = 128$. The results illustrate that if we expand the feature size, the encoder still extracts an accurate feature so that there is little impact on the performance of the policy; however, when we decrease the feature size, the encoder can hardly represent enough information in the latent space, so that the performance of the policy is diminished.

E. Real-World Deployment

The vision-based policy is deployed on a real-world race car to verify the algorithm's feasibility. The vehicle has an egocentric depth camera with a refreshing frequency of 30Hz. A visual-inertial state estimation [20] that also runs at 30Hz on the onboard computer produces the proprioceptive state used in the student policy in real time, and a VESC electronic speed controller serves as the low-level controller. Fig. 9(a) shows the software and hardware system architecture. The experimental site is a new $12m \times 6m$ racetrack, which cannot be seen in the training process. The racing trajectory of the vehicle with speed information in the real-world experiment is shown in Fig. 9(b). The recorded speed is generated by the visual-inertial state estimation. The vision-based policy achieves direct zero-shot sim-to-real transfer. The vehicle in the real-world experiment finishes the lap with a duration of $9.62s$ and an acceleration limit of $6m/s^2$. The maximum velocity on the lap reaches $5.58m/s$. The vision-based policy demonstrates robustness to depth images with random noise, as shown in Fig. 9(c), and smoothness of control commands sent to the VESC low-level controller, as shown in Fig. 9(d).

In order to display the generalization ability of our policy to different real-world scenarios, we evaluate the proposed student policy on two racetracks, as shown in Fig. 10. Two types of ground textures are laid on each track: PVC and asphalt. The results of running the student policy are shown in Table VII. Our policy can finish two racetracks on two different ground textures without any fine-tuning. Due to the lower friction coefficient of the PVC texture, the vehicle tends to understeer and drift when cornering. The trajectory is usually closer to the outside of the track, resulting in a longer lap time. Although the vehicle's speed is high when entering corners on the PVC ground, the vehicle's drift may lead to unsafe behaviors.

Furthermore, we also compare the racing performance of different control policies on two racetracks with asphalt

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

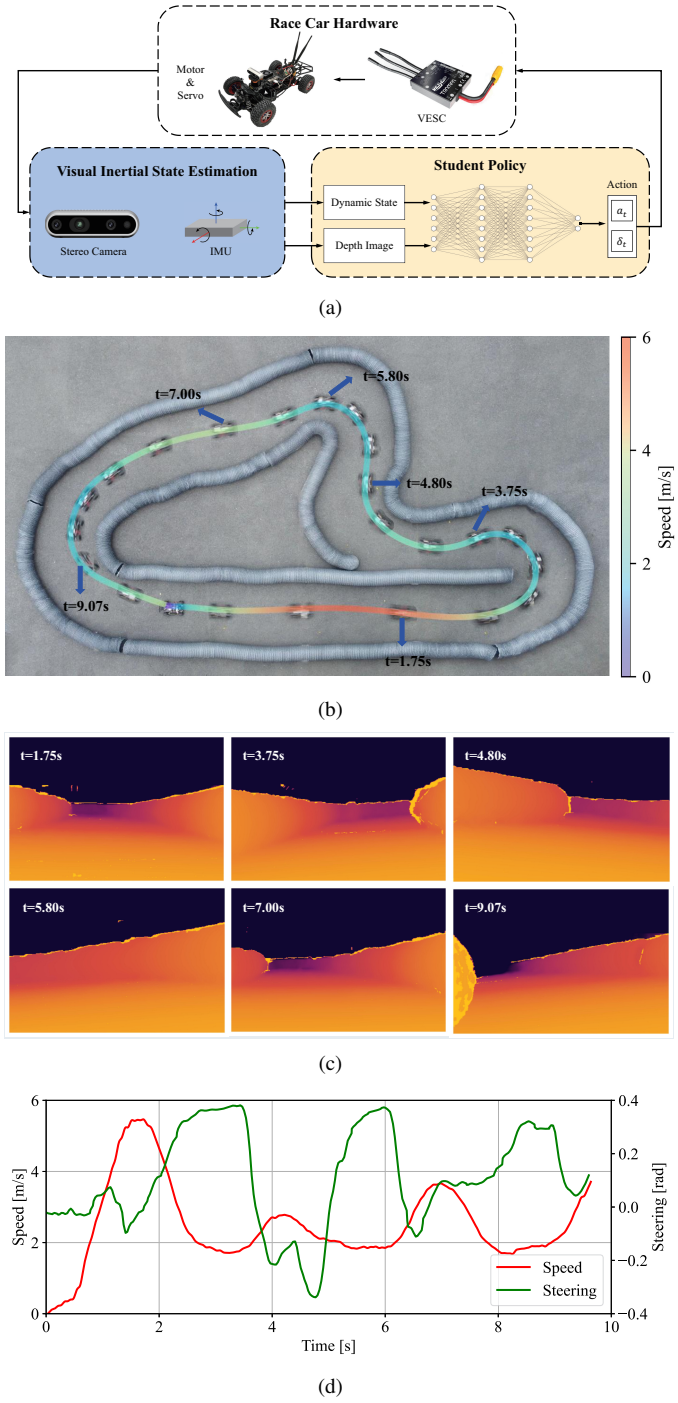
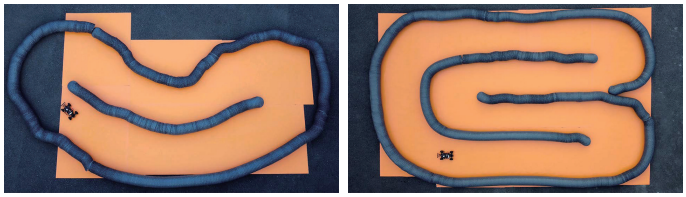


Fig. 9. Illustration of the end-to-end vision-based racing in a given racetrack ($12m \times 6m$), the proposed policy achieves zero-shot sim-to-real transfer. (a) The system used in the real-world experiment. (b) The racing trajectory in the real-world deployment with speed information, in which the maximum speed reaches $5.58m/s$. (c) The noisy depth images at several time steps from the front-facing camera, and the positions of the vehicle are labeled in (b). (d) The speed and the steering commands that were fed to the low-level controller in the real-world experiment.

textures. The results are shown in Table VIII. The student policy without a GRU component (Ours w/o GRU) displays the poorest generalization ability, which cannot finish any racetrack without a collision. The student policy without pre-trained VAE (Ours w/o pVAE) can finish Racetrack 1 and achieve a lap time similar to that of the proposed student

TABLE VII
REAL-WORLD EXPERIMENTAL RESULTS OF THE STUDENT POLICY EVALUATED ON TWO RACETRACKS WITH DIFFERENT GROUND TEXTURES.

Racetrack	Texture	Finished [✓or ✗]	Lap Time $[s]$	Max Speed $[m/s]$	Avg. Speed $[m/s]$
1	Asphalt	✓	6.96	4.37	2.81
	PVC	✓	7.23	4.26	2.76
2	Asphalt	✓	13.17	4.69	2.44
	PVC	✓	14.38	4.29	2.22



(a) Racetrack 1 ($7m \times 4m$) (b) Racetrack 2 ($8m \times 6m$)

Fig. 10. Two racetracks used in the real-world experiments, with the PVC texture.

policy, but cannot drive safely on Racetrack 2. The comparison between these two policies and the proposed student policy further validates the significance of observation memory and the robustness of the pre-trained encoder. Our student policy and the TAL policy exhibit the best generalization performance and finish the two racetracks. However, similar to the comparison in the simulation, our student policy achieves faster lap times than the TAL policy, demonstrating a better racing performance.

VI. CONCLUSION

This work presents a novel two-phase data-driven method to train an end-to-end policy to achieve high-speed vision-based autonomous racing. In the proposed algorithm, RL leverages privileged information to train a teacher policy, and a supervised learning algorithm distills optimal racing behaviors into a vision-based student policy. A VAE-RNN network architecture is designed in the student policy to process noisy depth images and figure out the memory of historical observations. Compared with model-based and learning-based baselines, the proposed method outperforms other methods in terms of three metrics: lap time, success rate, and mean jerk. The data scalability and generalization ability of our method are validated in the simulation experiments. An ablation study is also conducted to evaluate the impact of key components. Finally, the vision-based policy is deployed on a real-world race car, achieving zero-shot sim-to-real transfer.

Some limitations of the proposed method still require further study. First, our experiments were conducted in single-agent scenarios; the absence of multi-agent dynamics in our current policy could lead to suboptimal performance or safety issues in the competitive interactions with other vehicles. For instance, the policy may not overtake other vehicles to achieve a faster lap time or avoid dynamic obstacles. To address this, future work could explore multi-agent RL frameworks or game-theoretic approaches to enhance the policy's performance

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

TABLE VIII
REAL-WORLD EXPERIMENTAL COMPARISON RESULTS OF DIFFERENT CONTROL POLICIES EVALUATED ON TWO RACETRACKS.

Method	Racetrack 1				Racetrack 2			
	Finished [✓or ✗]	Lap Time $[s]$	Max Speed $[m/s]$	Avg. Speed $[m/s]$	Finished [✓or ✗]	Lap Time $[s]$	Max Speed $[m/s]$	Avg. Speed $[m/s]$
TAL	✓	9.04	3.59	2.47	✓	17.74	3.64	2.17
Ours	✓	6.96	4.37	2.81	✓	13.17	4.69	2.44
Ours w/o GRU	✗	-	5.17	3.17	✗	-	5.34	3.43
Ours w/o pVAE	✓	7.17	4.57	2.74	✗	-	5.59	3.49

in competitive racing scenarios. Another limitation is the method's reliance on detecting track boundaries through depth images. While effective in structured racetracks with clear boundaries, this approach may fail in off-road racing or tracks with ambiguous markings. In such environments, the depth camera may struggle to distinguish navigable paths, leading to suboptimal or unsafe behaviors. Future research could investigate alternative perception methods, such as semantic segmentation, to provide a more robust environmental understanding across diverse conditions. Sensor noise from depth cameras is another critical concern. Depth cameras are susceptible to artifacts caused by factors like reflective surfaces. Although our approach leverages domain randomization and denoising VAE during training, real-world noise can be more complex and unpredictable. Such noise could degrade the policy's decision-making, leading to failure to maintain optimal trajectories. To enhance robustness, future studies could develop a more realistic simulator for rendering images and explore advanced data augmentation techniques.

REFERENCES

[1] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 458–488, 2022.

[2] J. Betz, T. Betz, F. Fent, M. Geisslinger, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, M. Lienkamp *et al.*, "Tum autonomous motorsport: An autonomous racing software for the indy autonomous challenge," *J. Field Robot.*, vol. 40, no. 4, pp. 783–809, 2023.

[3] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale rc cars," *Optimal Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[4] J. Kabzan, M. I. Valls, V. J. Reijgwart, H. F. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan *et al.*, "Amz driverless: The full autonomous racing system," *J. Field Robot*, vol. 37, no. 7, pp. 1267–1294, 2020.

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[6] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," *Proc. Mach. Learn. Res.*, vol. 123, pp. 77–89, 2020.

[7] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs *et al.*, "Outracing champion gran turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[8] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürr, "Super-human performance in gran turismo sport using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4257–4264, 2021.

[9] 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.

[10] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 2070–2075.

[11] 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.

[12] J. Liu, H. Li, Z. Yang, S. Dang, and Z. Huang, "Deep dense network-based curriculum reinforcement learning for high-speed overtaking," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 1, pp. 453–466, 2022.

[13] J. Liu, Y. Cui, J. Duan, Z. Jiang, Z. Pan, K. Xu, and H. Li, "Reinforcement learning-based high-speed path following control for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 7603–7615, 2024.

[14] M. Vasco, T. Seno, K. Kawamoto, K. Subramanian, P. R. Wurman, and P. Stone, "A super-human vision-based reinforcement learning agent for autonomous racing in gran turismo," *arXiv:2406.12563*, 2024.

[15] Y. Song, K. Shi, R. Penicka, and D. Scaramuzza, "Learning perception-aware agile flight in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 1989–1995.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, 2014.

[18] C. Doersch, "Tutorial on variational autoencoders," *arXiv:1606.05908*, 2016.

[19] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist*, 2011, pp. 627–635.

[20] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.