

# A Champion-level Vision-based Reinforcement Learning Agent for Competitive Racing in Gran Turismo 7

Hoon Lee\*<sup>1</sup>, Takuma Seno\*<sup>2</sup>, Jun Jet Tai\*<sup>3</sup>, Kaushik Subramanian<sup>4</sup>,  
Kenta Kawamoto<sup>2</sup>, Peter Stone<sup>5,6</sup>, and Peter R. Wurman<sup>5</sup>

**Abstract**—Deep reinforcement learning has achieved superhuman racing performance in high-fidelity simulators like Gran Turismo 7 (GT7). It typically utilizes global features that require instrumentation external to a car, such as precise localization of agents and opponents, limiting real-world applicability. To address this limitation, we introduce a vision-based autonomous racing agent that relies solely on ego-centric camera views and onboard sensor data, eliminating the need for precise localization during inference. This agent employs an asymmetric actor-critic framework: the actor uses a recurrent neural network with the sensor data local to the car to retain track layouts and opponent positions, while the critic accesses the global features during training. Evaluated in GT7, our agent consistently outperforms GT7’s built-drivers. To our knowledge, this work presents the first vision-based autonomous racing agent to demonstrate champion-level performance in competitive racing scenarios.

**Index Terms**—Autonomous Agents, Reinforcement Learning, Vision-Based Navigation

## SUPPLEMENTARY VIDEOS

This paper is accompanied by a video of the performance: <https://youtu.be/cWqKbFsJcpo>

## I. INTRODUCTION

**A**UTONOMOUS racing demands self-driving vehicles to make split-second decisions at high speeds in dynamic, adversarial environments. Traditional control-based methods rely on modular pipelines for perception, planning, and control, requiring extensive hand-engineering. In contrast, deep reinforcement learning (RL) unifies these components, enabling end-to-end policy learning directly from sensor data. This approach has achieved superhuman performance in high-fidelity simulators like Gran Turismo 7 (GT7) through large-scale, distributed training [1].

Manuscript received: December, 18, 2024; Revised March, 20, 2025; Accepted April, 9, 2025.

This paper was recommended for publication by Markus Vincze upon evaluation of the Associate Editor and Reviewers’ comments.

\*These three authors contributed equally.

<sup>1</sup>Hoon Lee is with KAIST, Daejeon, South Korea.

<sup>2</sup>Takuma Seno and Kenta Kawamoto are with Sony AI, Tokyo, Japan. [takuma.seno@sony.com](mailto:takuma.seno@sony.com)

<sup>3</sup>Jun Jet Tai is with Coventry University, Coventry, UK.

<sup>4</sup>Kaushik Subramanian is with Sony AI, Zürich, Switzerland.

<sup>5</sup>Peter Stone and Peter R. Wurman are with Sony AI, New York, USA.

<sup>6</sup>Peter Stone is also with the University of Texas at Austin, USA.

Hoon Lee and Jun Jet Tai have worked on this project for their internships at Sony AI, Tokyo, Japan.

Digital Object Identifier (DOI): 10.1109/LRA.2025.3560873

©2026 IEEE

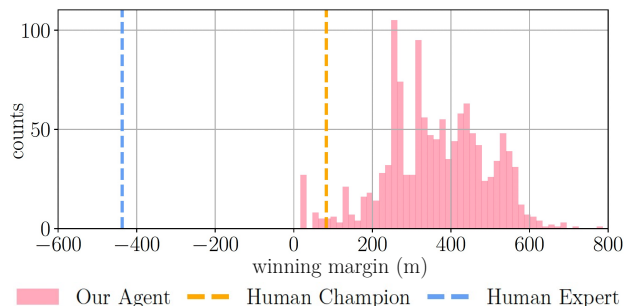
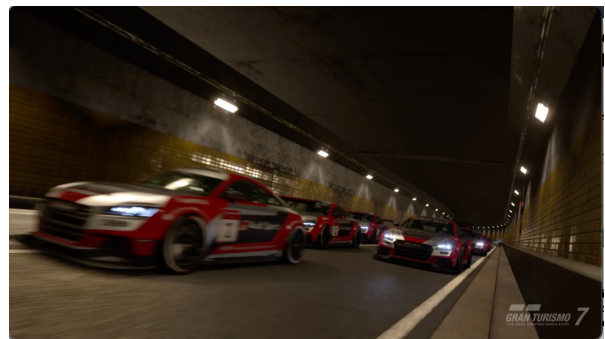


Fig. 1: **Top:** Our agent controlling an *Audi TT Cup* and racing against GT7’s built-in AI (BIAI). **Bottom:** Histogram of the *winning margin*, the distance between our agent and the leading BIAI at race completion. This evaluation involves starting from the last position versus 19 identical BIAI agents on the *Tokyo Expressway* track. Our agent consistently outperforms both Human Expert and Human Champion.

Despite these successes, transferring RL-based methods from simulation to real-world applications remains challenging. Current approaches rely on global features requiring external instrumentation, such as track geometry and opponent locations [1]–[7]. Acquiring accurate real-time global features is challenging and introduces latency, which impedes rapid decision-making essential for racing [8], [9]. This reliance on global features restricts the practicality of deep RL approaches in real-world racing scenarios.

A more feasible approach involves training agents using only onboard sensor data, such as ego-centric cameras and Inertial Measurement Units (IMUs), eliminating reliance on global features during inference. However using only on-board features can be challenging for competitive racing scenarios

that include partial observability due to occlusions and track layouts, in addition to the difficulty of handling high dimensional image data [10], [11].

Our work extends vision-based RL from time-trial [12] to competitive racing, where partial observability is more pronounced due to frequent occlusions of opponents and track layouts. We introduce an asymmetric recurrent actor-critic architecture [12], [13], where the actor relies on vision-based input with a recurrent memory module [14] to handle partial observability, while the critic leverages global state information during training. Additionally, to improve generalization and sample efficiency, we incorporate regularization techniques such as data augmentation [15], [16] and periodic network reinitialization [17], [18]. Finally, we include a multi-opponent racing reward function from Wurman et al [1] alongside a reward function for the time-trial agent [12].

We evaluate our agent in GT7, a high-fidelity racing simulator for PlayStation® 5. Trained against GT7’s built-in AI (BIAI), our agent consistently secures first place against 19 BIAIs, even when starting from the last position, outperforming human champions (Figure 1). Extensive ablation studies validate the effectiveness of the asymmetric architecture, recurrent memory module, and regularization strategies. To the best of our knowledge, this work presents the first vision-based autonomous racing agent to achieve champion-level performance in competitive racing scenarios [12], [19], [20].

## II. RELATED WORK

### A. Autonomous Racing

Autonomous racing aims to develop vehicles capable of performing at their dynamic limits in competitive environments [21], [22]. Traditionally, the problem has been divided into three main components: perception [23], [24], planning [25], [26], and control [27]–[31], with progress often occurring in isolation. Recently, reinforcement learning (RL) has emerged as a powerful tool for integrating these components into unified, end-to-end systems [1], [3]–[6]. For example, Fuchs et al. [3] achieved superhuman performance in a time trial racing scenario, where one car is on the track at a time, using a model-free RL approach with a novel reward structure. Subsequently, Wurman et al. [1] introduced Gran Turismo Sophy (GT Sophy), an RL agent that excels in both time trial races and multi-opponent races. However, during inference, these methods rely on global features, such as detailed track layout information and opponent localization, which are easily accessible in simulators but are challenging to obtain in real-world environments.

### B. Vision-Based RL for Autonomous Racing

Vision-based RL presents a promising alternative by enabling agents to operate competitively directly from visual inputs, eliminating the need for precise global features during inference. Despite its potential, existing methods face significant challenges. Jaritz et al. [32] reported that their vision-based RL agent struggled with maintaining optimal racing trajectories and frequently collided with obstacles. Cai et al. [19] combined imitation learning with model-based RL to

teach racing behaviors, but their approach required costly expert demonstrations, limiting its scalability. Additionally, many vision-based methods either lack direct comparisons to human drivers [4], [32] or fail to perform effectively in competitive settings [19], [20].

Vasco et al. [12] recently demonstrated that a vision-based agent could achieve superhuman performance in GT7. However, their work was confined to time trial settings without opponents, where partial observability and stochastic elements pose fewer challenges. In contrast, our work introduces the first vision-based RL agent to achieve champion-level performance in competitive racing scenarios, where the agent needs to interact with opponent cars and aim for the first position while respecting the rules of sportsmanship.

## III. METHOD

Our goal is to develop a vision-based agent for competitive racing scenarios in GT7 using only sensor data local to the car during inference. Our agent is built on top of the previous vision-based racing agent for time trial settings in GT7 [12].

### A. Observation Space

Our agent employs a multimodal observation space designed to capture the critical aspects of competitive racing. At each time step  $t$ , the composite observation  $\mathbf{o}_t = (\mathbf{o}_t^i, \mathbf{o}_t^p, \mathbf{o}_t^g)$  consists of image data  $\mathbf{o}_t^i$ , proprioceptive information  $\mathbf{o}_t^p$ , and global information  $\mathbf{o}_t^g$  derived from the GT7 simulation during training.

- **Image Feature** ( $\mathbf{o}_t^i$ ) is a  $64 \times 64$  RGB image, down-scaled from the original  $1920 \times 1080$  resolution. It captures the agent’s first-person view of the track. To simulate a front-view camera attached to the car, we disabled the in-game heads-up-display containing information about the vehicle speed or track map and masked out the rear-view mirror. Note that Vasco et al. [12] showed that the rear-view mirror was not critical for race car control.
- **Proprioceptive Feature** ( $\mathbf{o}_t^p$ ) includes data from the IMU sensors, defined as:

$$\mathbf{o}_t^p = [\mathbf{v}_t, \dot{\mathbf{v}}_t, \mathbf{v}_t^r, \mathbf{u}_t, \mathbf{s}_t, \mathbf{d}_t]$$

where  $\mathbf{v}_t \in \mathbb{R}^3$  represents the car’s linear velocity,  $\dot{\mathbf{v}}_t \in \mathbb{R}^3$  is the linear acceleration, and  $\mathbf{v}_t^r \in \mathbb{R}^3$  is the rotational velocity. The vector  $\mathbf{u}_t \in \mathbb{R}^3$  corresponds to the current inputs for steering, throttle, and brake. Lastly,  $\mathbf{s}_t \in \mathbb{R}^3$  and  $\mathbf{d}_t \in \mathbb{R}^3$  are the steering angle and changes in the steering angle over the last three time steps.

- **Global Feature** ( $\mathbf{o}_t^g$ ) includes track point information,  $c_t$ , and opponent grid data,  $g_t$ , proposed in Wurman et al. [1]. The track point feature consists of 177 3D coordinates  $c_t \in \mathbb{R}^{177 \times 3}$ , representing the edge of the track. These points are dynamically spaced based on the agent’s speed, covering approximately six seconds of travel time. The opponent grid feature contains information about nearby opponents. For each opponent, the position, velocity, and acceleration, projected onto the 2D plane, are recorded, forming a vector  $\mathbf{g}_t^{\text{opp}} \in \mathbb{R}^6$ . The full opponent grid feature

is represented as  $\mathbf{g}_t \in \mathbb{R}^{6 \times 14}$ , describing the 7 closest opponents looking ahead 75 meters ahead and 7 closest opponents looking behind 20 meters.

While the critic uses global features during training, they are excluded from the actor to ensure that the agent relies exclusively on information local to the car during inference.

### B. Action Space

We follow the action space used in previous work [12]:

$$\mathbf{a}_t = (a_t^s, a_t^g)$$

where  $a_t^s \in \mathbb{R}$  represents the *delta steering angle*, constrained to the range  $[-3^\circ, 3^\circ]$  to ensure realistic steering inputs. The term  $a_t^g$  denotes the *combined throttle and brake* value, within the range  $[-1, 1]$ , with  $-1$  indicating full braking and  $1$  indicating full throttle. Gear shifting is managed by the in-game automatic transmission system.

The agent's control updates occur at a frequency of 10 Hz, whereas the game operates at 60 Hz. To synchronize, the game applies a zero-order hold for throttle inputs and linearly interpolates the steering angle between control updates.

### C. Reward Function

We utilize a reward function used in previous work [12], incorporating a multi-opponent racing reward function from Wurman et al. [1], defined as a weighted combination of atomic reward components:

$$r_t = \lambda^p r_t^p + \lambda^o r_t^o + \lambda^b r_t^b + \lambda^v r_t^v + \lambda^c r_t^c + \lambda^s r_t^s + \lambda^t r_t^t + \lambda^h r_t^h.$$

- **Track Progress** ( $r^p$ ) measures the one-step change in the vehicle's track position since the last step. It is defined as  $r_t^p = p_t - p_{t-1}$ , where  $p_t$  represents the vehicle's position projected onto the closest point on the track center line.
- **Shortcut Penalty** ( $r^o$ ) penalizes the agent for taking shortcuts by cutting track corners. It is defined as  $r_t^o = -(s_t^o - s_{t-1}^o) |\mathbf{v}_t|$ , where  $s_t^o$  denotes the total time the vehicle has had at least three tires outside the track limits.
- **Barrier Collision Penalty** ( $r^b$ ) discourages the agent from using barrier collisions to change directions quickly. It is defined as  $r_t^b = -(s_t^b - s_{t-1}^b) |\mathbf{v}_t|$ , where  $s_t^b$  represents the total time the vehicle was in contact with a barrier.
- **Car Velocity-based Collision Penalty** ( $r^v$ ) penalizes the agent for colliding with other cars based on the difference in speed. It is defined as  $r_t^v = -|\Delta \mathbf{v}_t^x|^2$ , where  $\Delta \mathbf{v}_t^x$  is the speed difference between the agent's car and an opponent's car at the time of collision.
- **Car Collision Fixed Penalty** ( $r^c$ ) applies a constant penalty for any contact with opponent cars.
- **Overtaking Progress** ( $r^t$ ) rewards the agent for overtaking other cars. It is defined as  $r_t^t = \sum_{\forall i \in C \setminus k} [\mathbb{I}_{c_r < (p_t^i - p_t) < c_f} ((p_t - p_t^i) - (p_{t-1} - p_{t-1}^i))]$ , where  $p_t^i$  represents the position of car  $i$ ,  $k$  is the index of

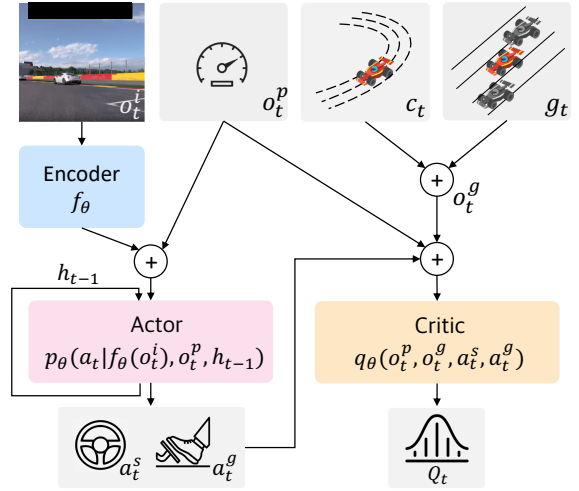


Fig. 2: **Architecture Overview.** The actor processes the image and proprioceptive features to predict actions, using a recurrent memory to track opponents and track layouts. The critic evaluates these actions using the global features. Both networks are jointly trained with the QR-SAC algorithm.

the ego-agent's vehicle, and  $c_r$  and  $c_f$  are thresholds for the minimum distance between two cars.

- **Steering Change Penalty** ( $r^s$ ) discourages abrupt steering changes. It is defined as  $r_t^s = -|\theta_t^s - \theta_{t-1}^s|$ , where  $\theta_t^s$  is the steering angle at time  $t$ .
- **Steering History Penalty** ( $r^h$ ) penalizes inconsistent steering decisions over a short period. It is defined as  $r_t^h = -m_t(1 + \exp(-c^s \cdot (\Delta_t - c^o)))$ , where  $\Delta_t = |\delta_t| + |\delta_{t-1}|$ ,  $\delta_t = \theta_t^s - \theta_{t-1}^s$  and  $m_t = \mathbb{I}_{\delta_t > c^d} \cdot \mathbb{I}_{\delta_{t-1} > c^d} \cdot \mathbb{I}_{\text{sgn}(\delta_t) \neq \text{sgn}(\delta_{t-1})}$ .  $c^s$ ,  $c^o$ , and  $c^d$  are constant factors.

Following previous work [1], [12], we used  $\lambda$  values as:  $\lambda^p = 1.0$ ,  $\lambda^o = 10.0$ ,  $\lambda^b = 20.0$ ,  $\lambda^v = 0.5$ ,  $\lambda^c = 6.0$ ,  $\lambda^s = 0.5$ ,  $\lambda^t = 3.0$ , and  $\lambda^h = 5.0$ . The constant factors are  $c_r = -20$ ,  $c_f = 40$ ,  $c^s = 182.883569$ ,  $c^o = 0.034$ , and  $c^d = 0.014$ .

### D. Architecture

We use Quantile Regression Soft Actor-Critic (QR-SAC) [33], a distributional variant of SAC [34] to train the agent. This algorithm has successfully learned superhuman autonomous racing agents in previous work [1], [12].

As illustrated in Figure 2, we use an *asymmetric* actor-critic architecture, which is designed as follows:

- **Actor** ( $\pi_\theta$ ): The actor is only provided with the local features, the image input  $o_t^i$  and proprioceptive data  $o_t^p$ . The image is passed through three convolutional layers,  $f_\theta$ . They use 32, 64, and 64 filters respectively. The kernel sizes are 8, 4, and 3 with strides of 4, 2, and 1. The resulting feature map is flattened and embedded into a 512-dimensional vector, which is then concatenated with the proprioceptive features. This combined feature vector is then processed by a recurrent predictor network,  $p_\theta$ , which includes an internal hidden state,  $h_{t-1}$ . The recurrent module is implemented

TABLE I: We evaluate our agents across three distinct scenarios, each consisting of a track, car, and tire combination.

Scenario	Track	Car	Tire
Tokyo	Tokyo Expressway - Central Clockwise, Japan	Audi TT Cup '16	Racing Hard
Spa	Circuit de Spa-Francorchamps, Belgium	Alfa Romeo 4C Launch Edition '14	Sports Medium
Sarthe	24 Heures du Mans race track, France	HYUNDAI N 2025 Vision Gran Turismo (Gr.1)	Racing Medium

with a Gated Recurrent Unit [35], followed by four fully connected layers, each with 2048 hidden units. A final linear layer with a hyperbolic tangent activation function predicts action probabilities of the *delta steering angle* and the *combined throttle and brake* value individually, modeled by a Gaussian distribution.

- **Critic** ( $q_\theta$ ): The critic uses both proprioceptive data  $\mathbf{o}_t^p$  and global features  $\mathbf{o}_t^g$  to precisely evaluate actions based on local and global information. The network consists of 4 fully connected layers with 2048 hidden units each and outputs a value function with 32 quantile units to model the Q-function distribution.

Using this asymmetric architecture, the actor relies on image and proprioceptive features, allowing the agent to make inferences based solely on local information.

#### E. Regularization

To improve the stability and generalization of our vision-based agent, we apply the following regularizations:

- **Network Reinitialization:** In RL, agents can overfit to early training data which often includes limited behaviors, such as navigating simpler track sections or less dynamic opponent interactions [18]. This can lead to overemphasis on static features like track layouts. To alleviate this bias, we reinitialize the networks after the replay buffer is fully populated, as recommended by Nikishin et al. [17]. At this stage, the buffer contains a diverse range of scenarios, including complex opponent interactions and strategic behaviors. Reinitializing the network allows the agent to restart from this broader dataset and helps to prevent the agent from prematurely overfitting to static features.
- **Image Augmentation:** To prevent overfitting to specific visual cues, we apply random shift augmentation [36]. During training, the input image is randomly shifted within a small range, simulating different visual perspectives and enhancing the agent's ability to generalize to unseen scenarios.

These regularization strategies improve the agent's learning stability and generalization, supporting robust decision-making in competitive racing environments.

## IV. EXPERIMENTAL SETUP

### A. Environments

We evaluate our approach in GT7 across three car-track scenarios, each presenting different challenges. Detailed setups are provided in Table I and Figure 3.

- **Tokyo:** A track with a mix of chicanes, high-speed straights, and tight track boundaries with no run-off, requiring precise

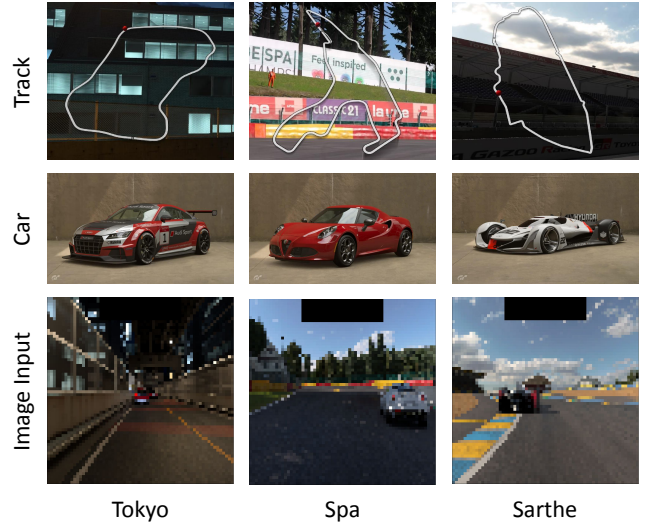


Fig. 3: **Racing Scenarios.** Visualization of track, car, and sample input image for each training scenario.

control in overtaking maneuvers with a front-wheel drive vehicle.

- **Spa:** A technical circuit with significant elevation changes, demanding good racing lines and control of vehicle oversteer with a rear-wheel drive vehicle.
- **Sarthe:** A high-speed circuit where effective slipstreaming on long straights and managing vehicle downforce are critical, with speeds reaching 340 km/h in a 4WD car.

### B. Training

Unlike many RL simulators (e.g., MuJoCo [37]) that can leverage accelerated simulation, GT7 operates in real-time. To speed up training, we used the same asynchronous distributed training framework described in Wurman et al. [1]. Our setup utilized 20 rollout workers for data collection, each connected via Ethernet to a dedicated PlayStation® 4 system.

Latency from retrieving images over Ethernet presented challenges for real-time training. To mitigate this problem, we configured the simulator to pause simulation steps until action commands were received from the rollout workers [12]. A dedicated training server managed the network parameters and updated them via gradient descent. Rollout workers are synchronized with the server at the end of each epoch by receiving the latest policy checkpoint.

While GT7 supports a maximum of 20 cars per track, training against 19 BIAI opponents directly can hinder the agent's ability to learn basic driving skills. To address this challenge, we adopted the multi-scenario training approach

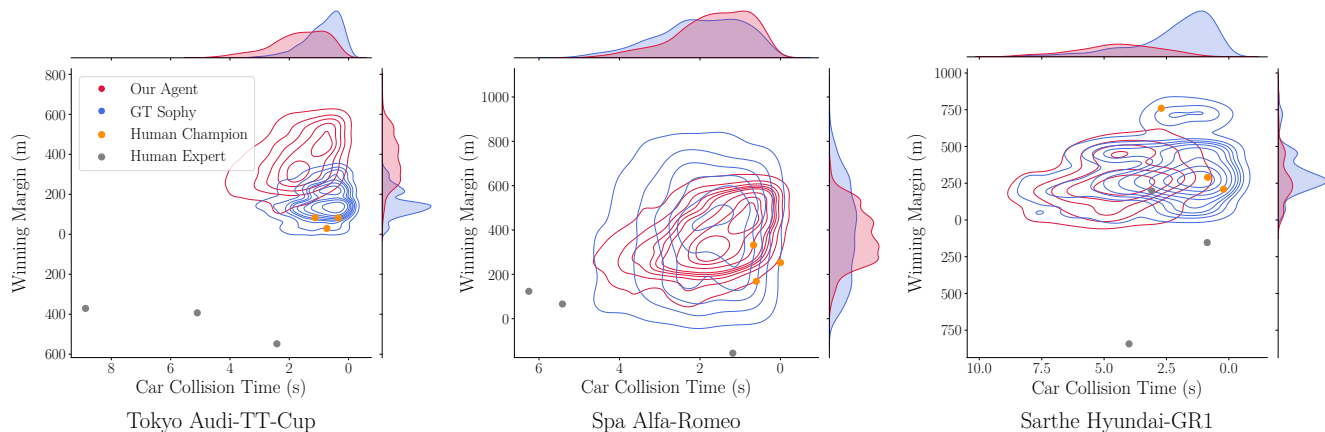


Fig. 4: **Performance comparison of our agent, GT Sophy, a Human Expert, and a Human Champion.** *Car collision time* is the total duration of contact with any opponent, while *winning margin* is the distance between the agent and the highest-ranked opponent when the agent completes all four laps. The contours represent the density of data points, with denser regions indicating more frequent occurrences of certain performance outcomes. Upper-right regions indicate superior performance, as they represent larger *winning margin* achieved with lower *car collision times*.

from Wurman et al. [1]. Each training episode is sampled from a range of configurations, from solo runs to races with 1, 2, 3, 4, 7, 12, and 19 opponents. The agent starts each race from randomly sampled points around the track. To diversify opponent behavior, GT7’s balance-of-performance (BoP) interface uniformly samples the opponent cars’ engine power and body weight within  $[-25\%, +25\%]$  range relative to vehicle’s original specification.

For the main experiments, we adopted the same hyperparameters as those established in prior work [12]. We used a mini-batch size of 512, sampled from 16 trajectories with a sequence length of 32 and a burn-in phase of 16 steps for the recurrent module. The replay buffer was set to 5 million samples, and we applied the Adam optimizer with a learning rate of  $2.5 \times 10^{-5}$ , a discount factor of 0.9896, and an entropy coefficient of 0.01. We incorporated a multi-step return with  $n = 7$ . To encourage the network to relearn overlooked features, we reinitialized the networks at 2,000 epochs, which aligned with the replay buffer reaching capacity. Image inputs were augmented using random shift with a maximum shift of 4 pixels, utilizing mirrored padding. The training was conducted over 20,000 epochs.

### C. Baselines

For comparison, we include the following baselines:

- **Human Expert:**<sup>1</sup> Performance of a GT7 player with 25+ years of experience in the Gran Turismo series and real-world circuit racing. The player is regularly ranked in the top 3-5% in online time trial events of GT7. The player was allowed unlimited practice laps and evaluated in three trials per scenario, with trials restarted if the player lost control and spun out to ensure consistency.

<sup>1</sup>We conducted evaluation trials with Rodney Meza to present results as a Human Expert. He is an employee of Sony Research, Tokyo.

- **Human Champion:**<sup>2</sup> Performance of a top GT7 player with multiple world titles. The evaluation protocol matched that of the Human Expert.
- **GT Sophy:** An agent trained using the architecture described in Wurman et al. [1], with modifications to train exclusively against BIAI. Population-based training is not utilized in this setup.

### D. Evaluation

We evaluated the performance of each agent over 4-lap episodes in their respective scenarios, starting from the back of a 20-car grid, with the remaining 19 cars controlled by BIAI. BIAI is an in-game model predictive control-based AI in GT7, serving as the opponent for all experiments. The BoP is disabled during evaluation. The primary metric was the *winning margin*, which measures the distance by which the agent leads the second-best opponent if it wins, or the deficit if it does not. Sportsmanship was also considered, with overtaking through collisions discouraged. We tracked *car collision time*, the total duration the agent’s car was in contact with others during an episode.

After training, we selected the top model checkpoint for each seed (three seeds in total) based on the highest *winning margin* with a *car collision time* better than the worst observed in the human champion baseline. The evaluation was conducted over 500 episodes per model checkpoint.

## V. EXPERIMENTAL RESULTS

### A. Main Experiment

We present the main experiment results in Figure 4, using a Kernel Density Estimate plot [38] to visualize the relationship between *car collision time* (x-axis) and *winning margin* at

<sup>2</sup>We conducted evaluation trials with Mikail Hizal, a champion at the GT World Series events in 2019 and 2020, to present results as a Human Champion.

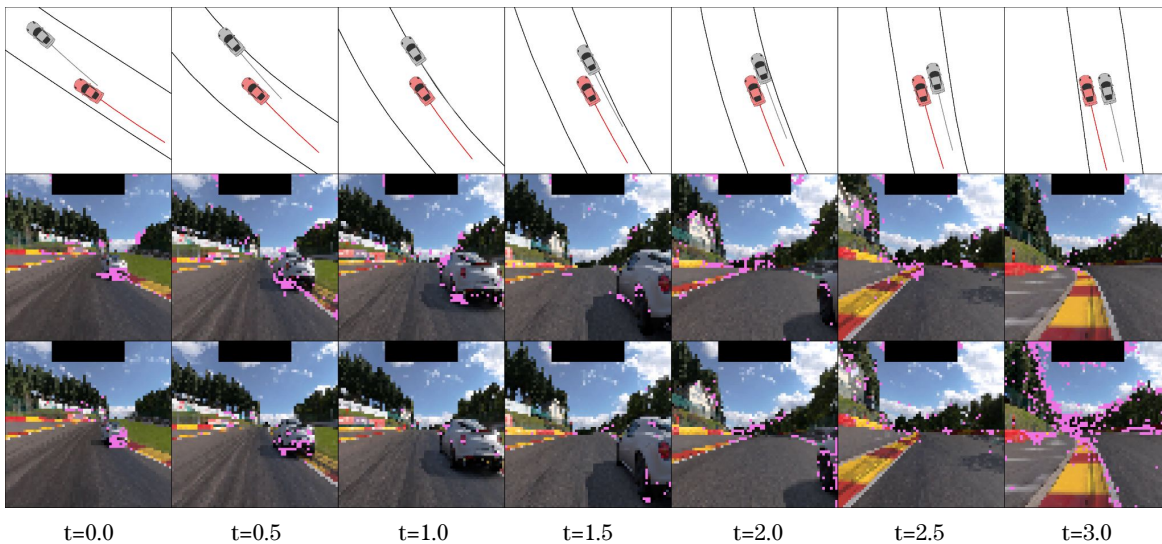


Fig. 5: **Visualizing our agent’s trajectory and action attributions in the Spa scenario.** The sequence is shown in 0.5-second intervals and consists of three rows: **Top:** displays the trajectory of our agent (red) and a BIAI opponent (black); **Middle:** shows attribution maps using Integrated Gradients, highlighting the agent’s focus on lower vehicle regions for overtaking opportunities or treelines for track layout. **Bottom:** illustrates how visual features from the past frames contribute to actions predicted for the final frame, demonstrating the agent’s ability to infer information that is not included in the final frame.

the final distance (y-axis). We focus on the 50% interquartile range, with better performance indicated by density concentrated on the upper-right corner (🚩). In all scenarios, our agent demonstrates champion-level performance:

- **Tokyo.** Our vision-based agent outperforms all baselines, achieving the highest *winning margin*. This superior performance likely stems from our agent’s ability to assess the distance and gaps to nearby cars better than GT Sophy. While GT Sophy treats opponents as point masses with relative position, velocity, and acceleration, it lacks awareness of opponent orientation [1]. In contrast, our agent’s vision-based input enables it to infer opponent orientation, leading to improved gap perception and overtaking capabilities. These advantages are particularly critical on the Tokyo track, which demands precise navigation within tight boundaries and minimal run-off areas.
- **Spa.** Our agent matches GT Sophy’s performance while surpassing the human expert and the human champion baseline. Overtaking is easier here due to the track’s wider layout and generous run-off areas, allowing our agent to exploit curbs and optimize its racing line. The similar performance between our agent and GT Sophy is expected, as both consistently execute near-perfect racing lines, providing a significant advantage over the human champion, whose racing lines are skilled but less precise.
- **Sarthe.** On Sarthe, our agent consistently surpasses the human expert and a majority of the human champion data in performance. Note that our agent induces higher *car collision time* compared to the human champion. Although we selected a model checkpoint with a collision time lower than the worst observed human performance, the agent’s overall collision time remains higher. This discrepancy is likely due to inherent randomness in GT7:

even with identical starting conditions, vehicle positions can diverge significantly after the first corner, where a majority of collisions occur. GT Sophy achieves a lower collision time than our agent, utilizing the precise perception to avoid collisions at the first corner.

### B. Visual Analysis

To better understand the decision-making process of our vision-based agent, we apply Integrated Gradients (IG) [39], which assigns importance to individual pixels in an image. In our analysis, attributions are calculated using a  $7 \times 7$  mean-filtered blurred baseline as a reference point. Gradients are then computed over 20 linearly interpolated images between the baseline and the input image, integrating the resulting gradients along this path. These integrated attributions are summed to quantify the contribution of each pixel to the output action. Figure 5 visualizes these attributions during a short driving segment, highlighting pixels contributing to the top 90% of attribution values. Past work has shown that pixels with high attributions are crucial to maintaining performance in the racing domain [12].

The middle row of Figure 5 shows that the agent exhibits context-dependent attention patterns. When near opponents, the agent focuses on lower vehicle regions and shadows to assess overtaking opportunities, similar to human drivers who rely on these cues in competitive driving [40]. On straight sections, the agent shifts attention to static features like vanishing points, treelines, and skylines, which aid in track localization and turn anticipation, reflecting the gaze patterns of professional drivers [41]. While lane markings occasionally attract attention, their reliability is influenced by lateral positioning and the presence of opponents. As a result,

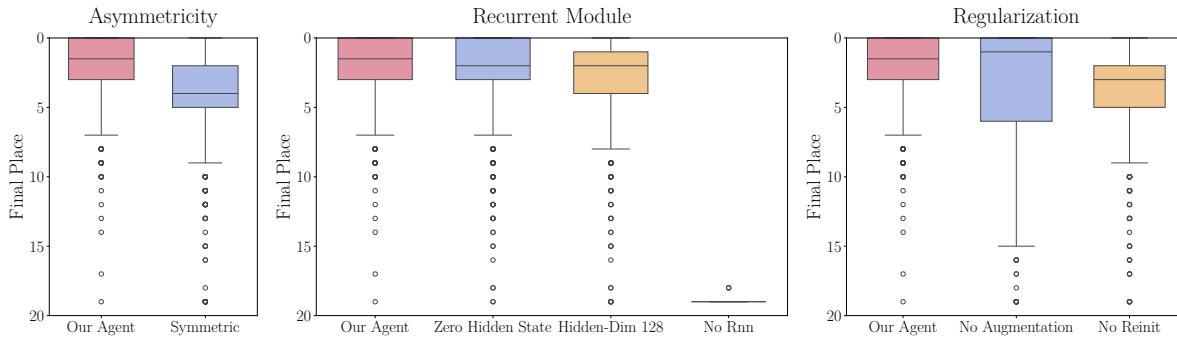


Fig. 6: **Ablation studies on the Tokyo scenario.** Ablated variations are compared by *final place* metric, evaluated over 500 episodes with three random seeds. **Left:** shows a comparison with the symmetric architecture variant; **Middle:** compares variations with different recurrent module configurations; **Right:** shows how each regularization approach affects performance.

the agent prioritizes more stable environmental cues, such as the sky and treelines.

The bottom row of Figure 5 illustrates how the agent utilizes the recurrent module to capture long-term dependencies. By performing backpropagation-through-time [42] to compute IG, we visualize how information from earlier frames contributes to actions predicted for the final frame in the sequence. This visualization demonstrates how the agent uses early-frame data to infer opponent positions and trajectories, which informs its decision-making for future maneuvers. This ability to integrate long-term predictions is crucial for partially observable, multi-player environments like racing, where opponent velocities cannot be directly inferred from a single frame.

### C. Ablation Studies

We conducted ablation studies to evaluate the contributions of specific architectural and training decisions to the performance of our vision-based agent. Each ablation involved modifications or removals of model components, followed by an evaluation to quantify their impact. We trained agents in each setting for 5,000 epochs and selected the top-performing checkpoint from three seeds for each setting based on the highest *winning margin*. Each model was evaluated over 500 episodes, and the results were summarized using box plots of each agent’s *final place*, the final place after the race.

- **Asymmetric Architecture:** The asymmetric architecture, where the critic incorporates both local (image and proprioceptive data) and global features, was pivotal for champion-level performance. In contrast, the symmetric variant, relying solely on local features, consistently failed to achieve first place in most evaluations. This result highlights the importance of leveraging global features in the critic.
- **Recurrent Module:** During training, initializing the recurrent module’s hidden state to zeros before RNN warmup slightly degraded performance compared to using the replay buffer’s stored hidden state. Reducing the hidden state dimension from 512 to 128 caused a noticeable performance drop, while completely removing the RNN resulted in complete failure, with the agent unable to overtake any opponents. These findings emphasize the RNN’s role in

maintaining temporal continuity, tracking off-screen opponents, and estimating their velocity and direction.

- **Regularization:** Applying image augmentation reduced the variance in performance across evaluation episodes, by potentially enhancing the agent’s generalization and mitigating overfitting to specific visual inputs. Similarly, reinitializing the networks at 2000 epochs improved stability and final performance, as they allowed the agent to relearn and emphasize underrepresented features in the replay buffer, leading to a more balanced use of diverse visual inputs during training.

## VI. CONCLUSION

In this work, we introduced a vision-based autonomous racing agent that achieves champion-level performance in Gran Turismo 7, using only ego-centric camera views and proprioceptive data for inference. Our approach leverages an asymmetric actor-critic framework, where the actor uses both ego-centric and proprioceptive inputs, enhanced by a recurrent neural network to capture track layouts and opponent dynamics. The critic, on the other hand, has privileged access to the detailed track and opponent information during training. The agent consistently outperformed model predictive control drivers, achieving overtaking maneuvers comparable to, or better than, those of human champions.

This work sets a new benchmark for vision-based competitive racing and demonstrates the potential of reinforcement learning in high-performance, real-time environments. However, our agent was tested in a controlled setting, using a single vehicle type per scenario with fixed weather conditions. Future work could focus on expanding the framework to enable competitive head-to-head races with top human drivers, as well as enhancing the agent’s ability to generalize across different car models, tracks, and weather conditions. Overcoming these challenges will be key to deploying vision-based racing agents in real-world scenarios.

## ACKNOWLEDGMENTS

We are very grateful to Polyphony Digital Inc. and Sony Interactive Entertainment for enabling this research. We would

also like to express our gratitude to Mikail Hizal, a GT champion, and Rodney Meza, an expert player in GT, for providing the reference data used in our evaluation.

## REFERENCES

- [1] 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.
- [2] A. Folkers, M. Rick, and C. Büskens, “Controlling an autonomous vehicle with deep reinforcement learning,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2025–2031.
- [3] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürri, “Superhuman performance in gran turismo sport using deep reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.
- [4] A. Remonda, S. Krebs, E. Veas, G. Luzhnica, and R. Kern, “Formula rl: Deep reinforcement learning for autonomous racing using telemetry data,” *arXiv preprint arXiv:2104.11106*, 2021.
- [5] Y. Song, H. Lin, E. Kaufmann, P. Dürri, and D. Scaramuzza, “Autonomous overtaking in gran turismo sport using curriculum reinforcement learning,” in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 9403–9409.
- [6] A. Remonda, N. Hansen, A. Raji, N. Musiu, M. Bertogna, E. Veas, and X. Wang, “A simulation benchmark for autonomous racing with large-scale human data,” *arXiv preprint arXiv:2407.16680*, 2024.
- [7] W. Xiao, H. Xue, T. Tao, D. Kalaria, J. M. Dolan, and G. Shi, “AnyCar to anywhere: Learning universal dynamics model for agile and adaptive mobility,” *arXiv preprint arXiv:2409.15783*, 2024.
- [8] I. A. Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, “A survey of state-of-the-art on visual slam,” *Expert Systems with Applications*, vol. 205, p. 117734, 2022.
- [9] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, “A comprehensive survey of visual slam algorithms,” *Robotics*, vol. 11, no. 1, p. 24, 2022.
- [10] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, “Learning latent dynamics for planning from pixels,” in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.
- [11] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, “Improving sample efficiency in model-free reinforcement learning from images,” in *Proceedings of the aaai conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 10674–10681.
- [12] 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 preprint arXiv:2406.12563*, 2024.
- [13] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, “Asymmetric actor critic for image-based robot learning,” *arXiv preprint arXiv:1710.06542*, 2017.
- [14] S. Kapturovski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney, “Recurrent experience replay in distributed reinforcement learning,” in *International conference on learning representations*, 2018.
- [15] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, “Reinforcement learning with augmented data,” *Advances in neural information processing systems*, vol. 33, pp. 19884–19895, 2020.
- [16] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, “Mastering visual continuous control: Improved data-augmented reinforcement learning,” *arXiv preprint arXiv:2107.09645*, 2021.
- [17] E. Nikishin, M. Schwarzer, P. D’Oro, P.-L. Bacon, and A. Courville, “The primacy bias in deep reinforcement learning,” in *International conference on machine learning*. PMLR, 2022, pp. 16828–16847.
- [18] H. Lee, H. Cho, H. Kim, D. Gwak, J. Kim, J. Choo, S.-Y. Yun, and C. Yun, “Plastic: Improving input and label plasticity for sample efficient reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [19] P. Cai, H. Wang, H. Huang, Y. Liu, and M. Liu, “Vision-based autonomous car racing using deep imitative reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7262–7269, 2021.
- [20] J. Herman, J. Francis, S. Ganju, B. Chen, A. Koul, A. Gupta, A. Skabelkin, I. Zhukov, M. Kumskey, and E. Nyberg, “Learn-to-race: A multimodal control environment for autonomous racing,” in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9793–9802.
- [21] P. Karle, M. Geisslinger, J. Betz, and M. Lienkamp, “Scenario understanding and motion prediction for autonomous vehicles—review and comparison,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16962–16982, 2022.
- [22] 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 Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.
- [23] F. Massa, L. Bonamini, A. Settini, L. Pallottino, and D. Caporale, “Lidar-based gnss denied localization for autonomous racing cars,” *Sensors*, vol. 20, no. 14, p. 3992, 2020.
- [24] W.-z. Peng, Y.-h. Ao, J.-h. He, and P.-f. Wang, “Vehicle odometry with camera-lidar-imu information fusion and factor-graph optimization,” *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–13, 2021.
- [25] T. Herrmann, F. Passigato, J. Betz, and M. Lienkamp, “Minimum race-time planning-strategy for an autonomous electric racecar,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.
- [26] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, “Optimization-based hierarchical motion planning for autonomous racing,” in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 2397–2403.
- [27] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. M. Rehg, and E. A. Theodorou, “Robust sampling based model predictive control with sparse objective information,” in *Robotics: Science and Systems*, vol. 14, 2018, p. 2018.
- [28] C. Hao, C. Tang, E. Bergkvist, C. Weaver, L. Sun, W. Zhan, and M. Tomizuka, “Outracing human racers with model-based autonomous racing,” *arXiv preprint arXiv:2211.09378*, 2022.
- [29] H. Xue, E. L. Zhu, J. M. Dolan, and F. Borrelli, “Learning model predictive control with error dynamics regression for autonomous racing,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13250–13256.
- [30] D. Kalaria, Q. Lin, and J. M. Dolan, “Adaptive planning and control with time-varying tire models for autonomous racing using extreme learning machine,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10443–10449.
- [31] A. Raji, A. Liniger, A. Giove, A. Toschi, N. Musiu, D. Morra, M. Verucchi, D. Caporale, and M. Bertogna, “Motion planning and control for multi vehicle autonomous racing at high speeds,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 2775–2782.
- [32] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, “End-to-end race driving with deep reinforcement learning,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2070–2075.
- [33] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, “Distributional reinforcement learning with quantile regression,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
- [34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.
- [35] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation. arxiv 2014,” *arXiv preprint arXiv:1406.1078*, 2020.
- [36] I. Kostrikov, D. Yarats, and R. Fergus, “Image augmentation is all you need: Regularizing deep reinforcement learning from pixels,” *arXiv preprint arXiv:2004.13649*, 2020.
- [37] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [38] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley-Interscience, 1992.
- [39] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.
- [40] V. Nagy, P. Földesi, and G. Istenes, “Area of interest tracking techniques for driving scenarios focusing on visual distraction detection,” *Applied Sciences*, vol. 14, no. 9, p. 3838, 2024.
- [41] P. M. Van Leeuwen, S. De Groot, R. Happee, and J. C. De Winter, “Differences between racing and non-racing drivers: A simulator study using eye-tracking,” *PLoS one*, vol. 12, no. 11, p. e0186871, 2017.
- [42] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.