Reinforcement Learning Based Control for Ego Vehicle Racing with Dynamic Opponents

Ruopei Chen
Mechanical Engineering
rpchen@stanford.edu

Yazhou Zhang
Mechanical Engineering
zhangyaz@stanford.edu

***Abstract*—This work investigates reinforcement learning for high-speed autonomous racing in a multi-agent environment with dynamic opponent vehicles. We develop a Proximal Policy Optimization (PPO) controller trained using a two-phase curriculum: first on a single-vehicle oval track to learn stable racing-line following and curvature-aware speed control, and subsequently in the presence of multiple kinematic opponents to acquire overtaking and collision-avoidance behaviors. The agent receives a compact ego-centered occupancy-grid observation and learns a continuous control policy for steering and longitudinal acceleration.**

Despite being trained exclusively on a single track, the resulting policy demonstrates strong zero-shot generalization across six unseen racetrack geometries, achieving competitive lap times relative to a classical PD baseline while maintaining stable high-speed performance. Qualitative analysis shows reliable centerline tracking, effective local evasive maneuvers, and robustness to stochastic road-friction noise. However, limitations arise in scenarios requiring anticipatory deceleration or strategic gap selection, where the agent often maintains full throttle and fails to slow down appropriately. These behaviors stem from the restricted perceptual range of the occupancy grid and the strong incentive toward continuous forward progress in the reward design.

Overall, this study highlights both the potential and limitations of PPO-based racing policies for continuous-control autonomous driving. The results motivate future work on safer reward structures, richer perceptual models, and hybrid control approaches integrating predictive safety constraints.

I. Introduction

High-performance autonomous racing requires an autonomous agent to make continuous, high-frequency decisions under uncertainty while operating near the physical limits of vehicle dynamics. Unlike conventional autonomous-driving tasks, racing emphasizes aggressive yet reliable control, demanding that the ego vehicle maintain high speed, follow a precise racing line, and respond intelligently to dynamic multi-agent interactions.

Recent research in autonomous racing has increasingly explored reinforcement learning (RL) as a means to achieve time-optimal and adaptive control in high-speed environments, building on the limitations of classical trajectory-tracking pipelines.

Early work such as TORCS-based [1] racing agents demonstrated the feasibility of using deep RL to learn end-to-end steering and throttle commands. They has shown that RL policies can not only complete full laps but also learn human-like, grip-limit driving behavior, including professional-level braking modulation and racing-line selection, using only visual inputs. This capability highlights the impact of RL as a powerful model-free alternative to traditional control pipelines. More recent studies have applied algorithms like PPO [2] and SAC [1], [3] to outperform PID or MPC baselines by discovering aggressive racing lines and data-driven cornering strategies. Multi-agent racing research [4] has further shown that RL policies can learn overtaking, blocking, and collision-avoidance behaviors, especially when trained with domain randomization or curriculum learning to handle uncertainty in opponent dynamics.

The objective of this project is to train a RL based ego agent to race on a single-lane track that is sufficiently wide to support parallel driving alongside other rule-based opponent vehicles. The environment is structured so that the ego agent must complete one lap while satisfying three primary objectives: 1) maintaining the racing line and staying on track, 2) minimizing lap time, and 3) avoiding collisions with other vehicles. The ego agent will initially be trained in a single-vehicle setting to learn efficient racing-line following and speed adaptation based on track curvature. The lap time achieved in this solo scenario will serve as the performance baseline. Subsequently, multiple rule-based opponent vehicles will be introduced to the environment, requiring the ego agent to develop overtaking strategies while continuing to minimize total lap time. Project success is defined as achieving comparable lap times to the baseline while safely navigating and overtaking opponent vehicles.

We summarize the organization of the report as follows. Section II describes the environment in detail, including its observation and action models, vehicle dynamics, stochastic transition processes, and reward construction. Section III outlines our methodological approach, beginning with a classical-control baseline used to validate the environment, followed by the reinforcement learning framework and training curriculum employed to develop the ego agent. Section IV presents empirical results across a range of racetrack configurations, and Section V offers a qualitative analysis of the agent's behavior, emphasizing both its strengths and observed failure modes. Section VI discusses the broader limitations of the current design and identifies several promising avenues for future extension. Finally, Section VII concludes the report.

II. Environment Setting

The environment used in this project is built on `HighwayEnv's RacetrackEnv`, modified to support continuous-control racing on a single wide lane with sufficient lateral space for overtaking [5]. Fig. 1 visualizes the race track configuration we used for training. The yellow rectangle is the ego agent, while the blue rectangles are opponent vehicles.

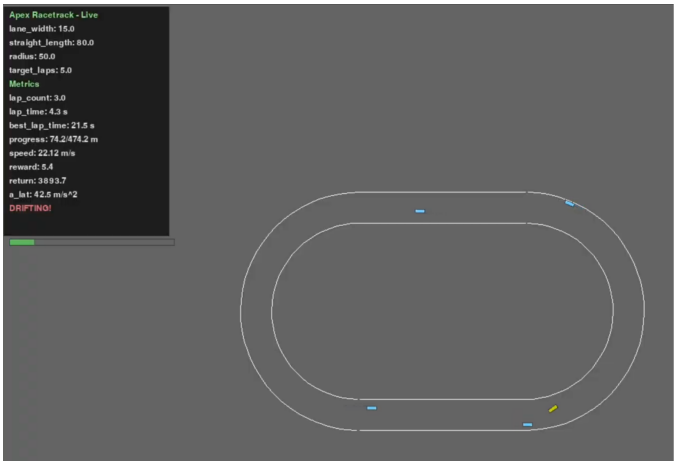

Fig. 1. Training race track configuration

A. State

The Markov state comprises the full simulator state: the road network (lane centerlines, widths, headings, speed limits), all dynamic vehicles (poses, headings, speeds, crash flags), static obstacles, and the controlled vehicle's kinodynamic state and lane index. The environment maintains additional latent race variables used for shaping and termination: absolute progress around the loop, per-step forward progress, per-lap progress, cumulative progress, current lap time, best lap time, lap count, track length, steering rate, and the most recently computed lateral acceleration. Although the simulator internally maintains this full state, it is not fully observable to the agent, making the task a POMDP.

B. Observation

At each control step, the agent receives an ego-aligned `OccupancyGrid` centered on the vehicle. The grid spans $[-6,\,42]\ \mathrm{m}$ longitudinally (behind/ahead) and $[-12,\,12]\ \mathrm{m}$ laterally (left/right), with a resolution of $3\,\mathrm{m}$. Each cell encodes two binary features: *presence*, indicating whether the cell is occupied by another vehicle, and *on-road*, indicating whether the cell lies on the drivable surface. The full grid is provided to the agent as a flattened vector.

This representation offers compact, egocentric spatial context about nearby agents and road geometry, while remaining invariant to the vehicle's global position and heading. Importantly, internal quantities such as lap count and progress along the track are not included in the observation. As a result, the agent must infer both surrounding traffic behavior and its own race progress solely from this short-range, locally aligned perceptual input.

C. Action

The agent outputs a 2-D continuous action $\mathbf{a}_t = [a_{\text{long}},\,\delta] \in [-1,1]^2$. The longitudinal acceleration (a_{long}) integrates to speed, while the steering component (δ) maps to the vehicle's front-wheel steering angle via the environment's action interface.

The environment clamps the action-implied speed range to $\pm 22\,\mathrm{m/s}$ at reset so that policy commands remain within the model's physical limits. The simulator runs at 30 Hz while the controller operates at 10 Hz, so each chosen action is applied for three simulation steps before a new control decision is made.

D. Transition Model

State transitions are governed by the vehicle dynamics, the traffic model, and additional modifications introduced in this environment. The ego vehicle follows a bicycle dynamic model that integrates the agent's acceleration and steering over the decision interval. A drift-governor mechanism monitors lateral acceleration and automatically introduces lateral accelerations, thereby modeling traction loss and enforcing realistic cornering behavior. To further emulate real-world variability, the environment injects stochastic road-friction (grip) noise into the dynamics, causing small random fluctuations in the achievable tire forces. This prevents overly brittle policies and encourages robustness to uncertain traction conditions.

Obstacle vehicles evolve according to the IDM traffic model and maintain a fixed lateral offset while progressing along their lane. After each update, the environment recomputes the ego vehicle's longitudinal progress along the track, updates its lap count, and checks for termination conditions such as off-road events or collisions. Except for the randomness introduced at reset and the stochastic grip noise, transitions are deterministic given the current state and action.

E. Reward Model

The reward function is designed to support time-optimal, smooth, and safe racing under partial observability. At each step, the agent receives several additive components:

Time penalty.: A negative reward proportional to the time elapsed (Δt) encourages faster lap completion, which means:

$$R_{\text{time}} = -\omega_{\text{time}}\,\Delta t \tag{1}$$

Dense progress.: To promote steady forward driving, the agent receives a reward proportional to its progress along the track (Δs) at each step. This gives

$$R_{\text{prog}} = \omega_{\text{prog}}\,\Delta s \tag{2}$$

Smoothness.: Penalties on both steering rate ($\frac{d\delta}{dt}$) and steering magnitude ($|\delta|$) promote smoother driving behavior by discouraging rapid oscillations and excessively large control inputs.

$$R_{\text{rate}} = -\omega_{\text{rate}}\left(\frac{d\delta}{dt}\right)^2 \Delta t \tag{3}$$

$$R_{\text{mag}} = -\omega_{\text{mag}}\,|\delta|\,\Delta t. \tag{4}$$

Straight-speed bonus.: The agent receives an additional reward for driving quickly when cornering forces ($a_{\text{lateral}}/a_{\text{lateral,max}}$) are low. This encourages stable straight-line acceleration.

$$R_{\text{straight}} = \omega_{\text{straight}}\, v \left(1 - \frac{a_{\text{lateral}}}{a_{\text{lateral,max}}}\right)_{+} \Delta t. \tag{5}$$

Safety penalties.: Off-road behavior accrues a penalty, while collisions incur a fixed negative cost and terminate the episode.

$$R_{\text{off\_road}} = -\omega_{\text{off\_road}} \Delta t \tag{6}$$

$$R_{\text{collision}} = -\omega_{\text{collision}} \tag{7}$$

Lap completion.: A large terminal reward is granted when a lap is completed.

$$R_{\text{completion}} = 100 \tag{8}$$

Together, these components balance speed, stability, and safety, guiding the agent toward robust racing behavior in a stochastic and partially observable environment. In addition, the reward components and their associated weights are iteratively tuned across several training iterations, resulting in a configuration that consistently yields robust policy behavior.

III. Methods

A. Classical PD Control Baseline

We started by implementing a classical-control baseline to validate the environment's mechanics, including environment metrics, lap and progress computation, and drift governance. The controller combines PD lane-centering for steering with curvature-aware speed targeting for longitudinal control.

At each step, the ego pose is projected onto the reference lane to obtain longitudinal–lateral coordinates (s, r) and heading error ψ. The steering command is

$$\delta = -k_r \left(\frac{r}{w}\right) - k_\psi \psi \tag{9}$$

where w is the lane width. The command is subsequently clipped to the steering limits.

For speed control, curvature (κ) is computed from the bicycle model (see Eq. 10).

$$\kappa = \frac{2 \sin \beta}{L} \tag{10}$$

where L is the distance between the vehicle's front and rear axles, and β is the steering geometry angle which can be related to δ:

$$\beta = \arctan\big(0.5 \tan \delta\big) \tag{11}$$

The curvature-feasible speed, v_κ, is

$$v_\kappa = \sqrt{\frac{a_{\text{lateral,max}}}{|\kappa|}} \tag{12}$$

and the desired speed (v_{des}) is cliped to v_{cap}

$$v_{\text{des}} = \min(v_{\text{cap}},\, v_\kappa) \tag{13}$$

A proportional controller produces the longitudinal action:

$$a_{\text{long}} = k_v (v_{\text{des}} - v) \tag{14}$$

The baseline operates at 10 Hz control atop a 30 Hz simulation loop, yielding stable lane-following behavior. While this controller achieves reliable path tracking on an empty track, it cannot react to moving vehicles without substantial additional control modules, such as gap-selection rules, collision-avoidance heuristics, or an MPC formulation [6]. This motivates the use of learning-based policies capable of reasoning about multi-agent dynamics directly from perceptual input.

B. PPO controller

When choosing a reinforcement learning framework, we first excluded classical value-based methods such as Q-learning and DQN-style algorithms. The environment exhibits high-dimensional observations, a continuous action space, and long-horizon decision requirements under partial observability. Discretizing the control inputs would substantially reduce actuation fidelity, while function-approximation variants of Q-learning [7] are known to be difficult to stabilize in continuous-control domains.

We also considered alternative RL algorithms such as SAC and TD3, but these methods introduce additional complexities, including stochastic policy entropy tuning (SAC) or sensitivity to target smoothing and noise parameters (TD3), that can produce overly aggressive behaviors in safety-critical racing scenarios.

For these reasons, we adopt Proximal Policy Optimization (PPO) as our learning algorithm, using the Stable-Baselines3 implementation for reliable on-policy training and built-in tooling such as vectorized environments, callbacks, and TensorBoard logging. PPO [8] optimizes a stochastic policy using clipped surrogate objectives and generalized advantage estimation (GAE), which we found to be robust under continuous control and partial observability. Training was organized into stages. In Phase 1, we trained exclusively on the oval track (race track configuration shown in Fig 1) without moving obstacles to refine reward shaping and establish basic speed and trajectory behaviors. This curriculum enabled the policy to learn effective full-throttle straight-line driving, stable corner-entry behavior, and consistent lane-center tracking within roughly one hour of training. The resulting center-seeking behavior also improved robustness: when random slippage induced lateral drift, the policy retained sufficient margin to recover to the preferred path rather than compounding errors.

In Phase 2 we introduced moving traffic and shaped the agent's interactions with it. We began with two constant-speed obstacle vehicles placed on the lane centerline, but the agent quickly exploited this setup by hugging the outer boundary to bypass them. To eliminate this bias, we randomized obstacle spawns each episode—varying their longitudinal position, speed, and fixed lateral offset across the lane width. We then increased the number of vehicles and lowered their speeds to create larger speed differentials, forcing the agent to perform

earlier and more decisive evasive maneuvers with reduced reaction time. All obstacle vehicles were kept kinematic (non-reactive), ensuring that they appeared consistently in the OccupancyGrid's presence and velocity channels without coupling their behavior to the ego agent.

Although the policy was trained solely on the oval map, it demonstrated strong adaptability at test time, successfully transferring to several previously unseen track geometries without any additional fine-tuning. We next evaluate the learned policy across this suite of tracks to assess its generalization, robustness, and interaction behavior under dynamic traffic.

IV. Results

Phase 1 and Phase 2 required approximately 1 hour and 10 hours of training on an Apple M4 GPU, respectively. Fig. 2 shows the evaluation reward curve for Phase 2, which rises quickly and reaches a stable plateau after roughly 4 hours of training. The policy at this plateau is the one used for all subsequent experiments.

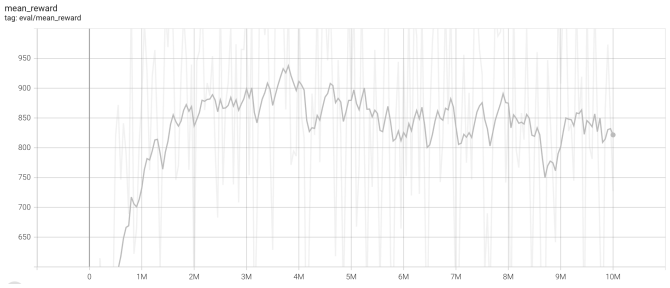

Fig. 2. Different race track configurations

Fig. 3 visualizes 6 different race track configurations that we used during the inference time. Tracks (a)–(d) are oval-shaped layouts with varying turn radii and straight-segment lengths, resulting in different curvature profiles and difficulty levels. Track (a) is the same track we used during the training time, and all other configurations are unseen by the policy learned. The track (b) has more opponents vehicles than other tracks. Tracks (e) and (f) introduce asymmetric shapes with sharper corners and elongated sections, creating more complex driving dynamics and requiring stronger generalization of the policy.

Table I reports the best lap times achieved by the ego agent, both with and without opponent vehicles, across the racetrack configurations evaluated during inference. The performance of a classical PD controller is also provided for comparison. Fig. 4 visualizes the differences between the three policies. Across all tracks, the PPO agent consistently maintains competitive performance relative to the PD controller, while exhibiting stronger robustness when opponents are present. The variation in results across track geometries further highlights the agent's ability to generalize to unseen layouts and traffic conditions. Video demonstrations of these runs are available at https://drive.google.com/drive/folders/1_XOykjNOtpOrx7_2qxZIqQGxwDtT1TW?usp=drive_link.

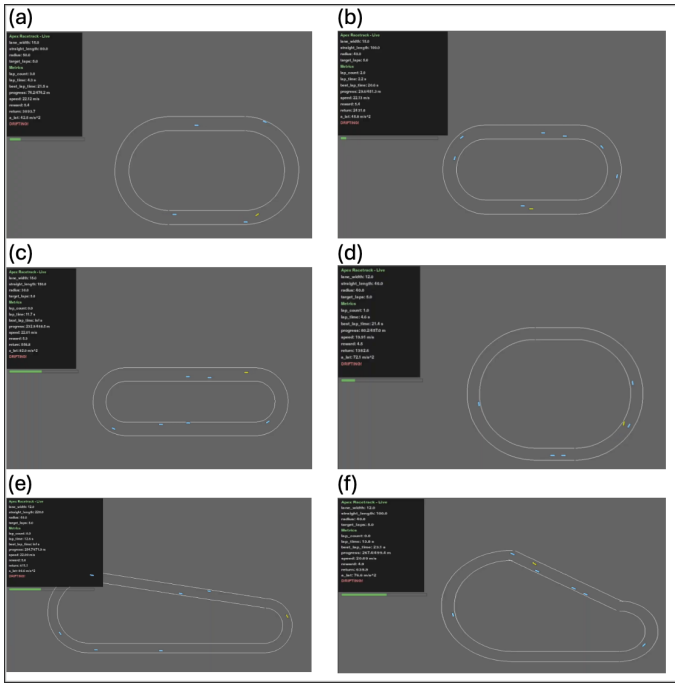

Fig. 3. Different race track configurations

TABLE I
BEST LAP TIMES FOR THE EGO AGENT ACROSS SIX RACETRACK CONFIGURATIONS. PPO RESULTS ARE SHOWN UNDER "NO OPPONENT" AND "WITH OPPONENTS," WITH A CLASSICAL PD CONTROLLER INCLUDED AS A BASELINE.

Track	No Opponent	With Opponents	PD Controller
a	18.9 s	21.5 s	20.2 s
b	20.5 s	21.1 s	21.7 s
c	22.2 s	22.9 s	23.9 s
d	15.0 s	21.5 s	17.7 s
e	29.9 s	30.7 s	30.2 s
f	21.6 s	23.1 s	22.9 s

V. Discussion

Our results indicate that the PPO-trained policy can generalize beyond the training distribution, demonstrating stable lane following and effective evasive maneuvers across a diverse set of track geometries. Qualitatively, the agent exhibits smooth corner entry, consistent center-seeking behavior, and reasonable traffic avoidance despite being trained exclusively on a single oval track. Across the evaluation suite, the learned policy maintained high-speed driving, adapted to new layouts without fine-tuning, and executed effective local steering adjustments in response to road-friction noise and nearby obstacles.

On Track a, the agent adopted a reliable centerline-following strategy, steering primarily to counteract the stochastic slippage introduced by the friction noise model. When encountering moving vehicles, the agent maintained near-maximum speed while shifting laterally to avoid collisions, indicating that the policy learned lightweight evasive behaviors without explicit trajectory planning. Track b increased the difficulty

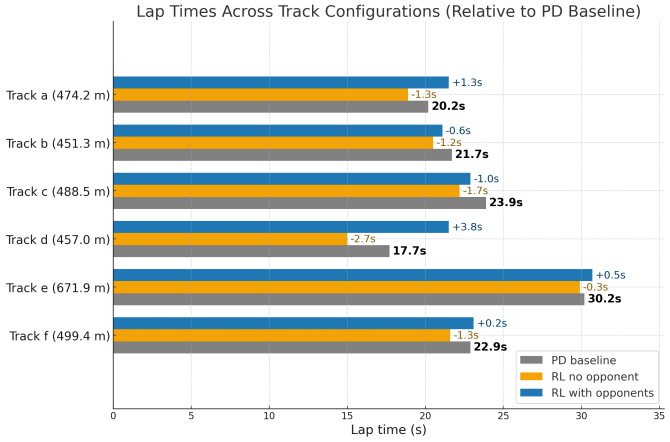

Fig. 4. Lap time comparison across track configurations, showing PD baseline performance and relative gains of RL agents with and without opponents.

by doubling the number of opponent vehicles and requiring completion of five consecutive laps. Despite this denser multi-agent setting, the agent completed the task consistently while maintaining high speeds, suggesting effective robustness under sustained interactions.

Tracks c and d modified the geometric properties of the course by varying straight lengths and corner radii. The agent navigated these layouts successfully, but limitations emerged under more extreme curvature. When the corner radius was reduced below approximately $25\,\mathrm{m}$, the policy frequently failed to slow down and instead attempted to maintain full throttle through the turn, resulting in loss of control. This highlights a broader limitation: the agent rarely learned to brake strategically, likely because optimal behavior in the training environment favored near-constant full-throttle driving. This effect is further amplified by the relatively high weight assigned to the dense progress reward (Eq. 2) at each step. Since maximizing Δs directly incentivizes continuous forward motion, the policy tends to prioritize speed over cornering safety and does not develop anticipatory braking behaviors. Shorter straightaways also made overtaking more challenging. Because overtakes often had to occur mid-corner, the agent was already steering near saturation, and additional steering required for obstacle avoidance increased the likelihood of drifting outside track boundaries.

Tracks e and f introduced asymmetric geometries with slanted straights and more complex corner sequences.The same policy, trained solely on the oval, was able to complete both tracks and overtake moving vehicles in varied contexts, demonstrating nontrivial zero-shot generalization. However, occasional abnormal behaviors were observed: in some rollouts, the agent failed to react adequately to slower vehicles and collided despite having available maneuvering space. We attribute these failures largely to limitations in the OccupancyGrid representation. With a restricted perceptual range, the agent has limited foresight regarding upcoming traffic and boundary geometry, reducing its ability to plan evasive actions with sufficient lead time. Expanding the observation radius or incorporating richer sensory modalities could improve anticipation and reduce such collision modes.

Overall, the results show that PPO can learn a high-speed racing policy that generalizes to new track layouts and multi-agent scenarios, but also reveal structural limitations arising from the reward design and observation model that affect braking behavior, overtaking reliability, and long-horizon anticipation under more complex geometries.

VI. Limitations and Future Work

However, several limitations remain. The obstacle vehicles in our environment are kinematic and do not respond to the ego vehicle, which simplifies multi-agent interactions and prevents the development of more realistic reciprocal behaviors. The observation model is based on a short-range occupancy grid rather than raw vision, restricting the agent's perceptual field and limiting applicability to real sensor pipelines. Additionally, the agent often failed to slow down in scenarios where safe overtaking required temporary deceleration, such as when gaps between opponent vehicles were too narrow or when entering tight corners. This behavior stems in part from the dense progress reward, which strongly incentivizes continuous forward motion, and from the limited ability of the observation model to anticipate upcoming hazards. Finally, our training curriculum relied on a single base track; although the agent demonstrated zero-shot generalization, broader environment randomization may be necessary to develop more robust and safety-aware policies.

Future work should therefore explore more realistic and interactive traffic models, including reactive or multi-agent opponents capable of negotiating space with the ego vehicle. Richer forms of domain randomization, such as varying friction conditions, lane widths, and traffic densities, may further strengthen generalization and reduce failure modes stemming from distributional shift. Expanding the observation space to vision-based modalities or enlarging the perceptual range could enable more anticipatory decision-making, improving the agent's ability to slow down, plan overtakes, and avoid collisions. Additionally, hybrid control architectures that combine model predictive control (MPC) with RL may offer improved safety guarantees by providing mechanisms for long-horizon planning and constraint enforcement. Investigating these directions would substantially enhance both the robustness and safety of autonomous racing agents in complex multi-agent environments.

VII. Conclusion

In this work, we developed and evaluated a reinforcement learning–based control policy for high-speed autonomous racing in the presence of dynamic opponent vehicles. Using a two-phase PPO training curriculum, the agent first learned robust racing-line following and speed control on a simplified single-vehicle track, and subsequently acquired evasive maneuvers and overtaking behaviors when interacting with

multiple rule-based opponents. Despite being trained exclusively on a single oval layout, the resulting policy exhibited strong zero-shot generalization, successfully navigating several unseen racetrack geometries with different curvature profiles, straight-line lengths, and opponent densities. Quantitatively, the agent achieved competitive lap times relative to a classical PD baseline and maintained stable, high-speed performance across diverse test conditions.

However, our findings also reveal several structural limitations of the current approach. The agent frequently maintained full throttle even in situations where strategic deceleration would improve safety or stability. This manifested most clearly when corner radii became tighter than those encountered during training, or when the spatial gap between opponent vehicles was insufficient for a safe overtake. In such cases, an optimal strategy would involve temporarily slowing down to wait for a viable passing opportunity, yet the learned policy rarely adopted this behavior. We attribute this to both the limited perceptual range of the OccupancyGrid representation and the strong incentive toward continuous forward progress imposed by the dense progress reward. Addressing these issues may require revised reward shaping, larger perceptual fields, or explicit mechanisms for modeling safe-gap reasoning and anticipatory deceleration.

Overall, this project demonstrates that PPO can serve as a reliable and adaptable control strategy for continuous-control autonomous racing, offering both competitive performance and meaningful generalization. Future extensions incorporating reactive opponents, broader environment randomization, or hybrid model-based/model-free architectures may further enhance robustness, safety, and strategic decision-making in complex multi-agent racing scenarios.

REFERENCES

[1] G. Bári and L. Palkovics, "Vision based driving agent for race car simulation environments," *arXiv preprint arXiv:2504.10266*, 2025.

[2] M. S. Holubar and M. A. Wiering, "Continuous-action reinforcement learning for playing racing games: comparing SPG to PPO," *arXiv preprint arXiv:2001.05270*, 2020.

[3] F. Tong, R. Liu, G. Yin, S. Zhang, and W. Zhuang, "Multi-policy Soft Actor-Critic reinforcement learning for autonomous racing," in *Proc. 2024 IEEE 18th International Conference on Advanced Motion Control (AMC)*, 2024, pp. 1–7.

[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*, 2022.

[5] E. Leurent, *An Environment for Autonomous Driving Decision-Making*, GitHub repository, 2018. Available: https://github.com/eleurent/highway-env

[6] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *arXiv preprint arXiv:1604.07446*, 2016.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2019.

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