2024 IEEE Intelligent Vehicles Symposium (IV)
June 2-5, 2024. Jeju Island, Korea

A LiDAR-based approach to autonomous racing with model-free reinforcement learning

Máté Hell, Gergely Hajgató, Ármin Bogár-Németh, Gergely Bári

2024 IEEE Intelligent Vehicle Symposium (IV) | 979-8-3503-4881-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/IV55156.2024.10588613

Abstract— This paper explores the use of reinforcement learning (RL) in the context of autonomous vehicle racing, specifically focusing on the F1TENTH simulation platform. While commercial autonomous driving often employs classic control algorithms, the state-of-the-art solutions, including those in the F1TENTH domain, increasingly rely on RL. Notably, RL-based approaches have shown superhuman performance in simulated environments, as seen in drone racing and the recent achievement by Sony in autonomous racing. In this paper we propose a novel LiDAR-only observation for learning vehicle dynamics, and test it with a widely accessible model-free RL method. The trained agent demonstrates the capability to transfer its driving skills to previously unseen tracks. Additionally, the paper provides recommendations for selecting hyperparameters, contributing valuable insights for newcomers to the field of autonomous racing.

I. Introduction and related work

Autonomous vehicle racing gained attention in the recent years as it provides a challenging, entertaining, yet safe environment for autonomous driving development. Like in human-driven vehicle racing, autonomous racecars are operated on closed racetracks with such safety measures that prevents contact with living creatures even in worst-case scenarios. This property is inevitable for the development of self-driving algorithms in their early stages; hence, many of the cutting-edge research are tested on autonomous racecars instead of self-driving cars in the everyday traffic.

The spectrum of available autonomous vehicle platforms for experimenting is wide, including small-scale, simplified and cost-effective vehicles on the one end (e.g. Duckietown [1]), and real racecars or passenger cars equipped with autonomous stack on the other end (e.g. the DARPA Grand Challenge [2] or the Indy Autonomous Challenge [3]). The F1TENTH platform [4] lies in the middle of this spectrum providing a compromise between model complexity and costs. F1TENTH cars are down-scaled, electric-driven vehicles with Ackermann-steering and double-fishbone suspension mimicking the mechanics of a real-sized racecar. Beside the physical car, the F1TENTH ecosystem contains a simulator [5] to foster autonomous driving software development.

While commercial autonomous driving use classic control algorithms (see [6], [7] among others), many of the state-of-the-art (SOTA) solutions are based on reinforcement learning (RL). A notable result is presented recently by the researchers of Sony, who trained a RL-agent for autonomous racing in a simulated environment and achieved superhuman performance in terms of laptime [8], [9]. The agent was not tested on a physical car, leaving the questions about the interpretability of the algorithm to a real environment open. However, competitive results based on RL can be found for both the simulated and the real world in such domains, where the physical equipment is relatively cheap and can be operated in a safe environment. One of these domains is drone racing, where Kaufmann et al. [10] achieved superhuman performance with a RL-agent recently.

Besides, RL-based solutions are emerging in the F1TENTH scene as well. Evans et al. [11] investigated the effect of the reward function to the agent's behavior in the presence of obstacles on the racetrack. Among three reward signals the fastest laptime was achieved with a reward based on the vehicle's progress on the track combined with the punishment of steering. The experiments were performed with hybrid planning introduced in [12], where the vehicle's trajectory is defined by a global trajectory optimizer algorithm which trajectory is modified with an RL agent in need.

The same authors compared three control architectures, both involving RL on a different level [13]. In two of the architectures, the localization and the trajectory planning were carried out with classic solutions. The third architecture involved end-to-end RL letting the agent make control decisions based directly on the sensory data. While the former two methods performed better in simulations, the end-to-end solution behaved better on the real F1TENTH racecar. This finding is in line with the following works where different aspects of end-to-end RL were examined with testing both in the simulation and the real F1TENTH environment.

Brunnbauer et al. [14] investigated the applicability of model-free and model-based RL-agents for self-driving. They found that Dreamer – a model-based method introduced in [15], [16] – outperforms SOTA model-free methods not only in terms laptimes but in knowledge transfer as well. This conclusion was drawn from experiments where the trained agents were evaluated on racetracks that were not seen by the agent during training. While the agent based on Dreamer was able to achieve a competitive laptime on unseen tracks, model-free RL solutions could not finish a complete lap in most

The research was supported by the National Research, Development and Innovation Office. (2020-2.1.1-ED-2021-00162)

All Authors with Széchenyi István University, H-9026 Győr, Hungary (e-mail: [mate.hell,gergely.hajgato,armin.bogar-nemeth,gergely.bari]@humda.hu])

of the cases. Model-free methods achieved an inferior performance despite the fact that their hyperparameters were optimized prior the final training.

Bosello et al. [17] investigated the ability of knowledge transfer with a less popular, yet robust RL-technique, namely deep Q-networks (DQNs). While the approach demanded on the great simplification of control possibilities – as the algorithm can handle only a discrete action space – the results were still satisfying. The training were more sample-efficient compared to not only the SOTA model-free methods but to the Dreamer-based model proposed by Brunnbauer et al. [14]. Moreover, the authors observed that the agent's knowledge is transferable, achieving better laptimes on unseen racetracks than previously reported by Brunnbauer et al. [14].

However, the proposed agent works with different observations, i.e., they make control decisions based on different representations of the vehicle's state. The DQN agent utilizes the LiDAR scan and the vehicle speed as the addition of the latter found to be enhance the model performance by the authors. Beside the momentarily values, the values corresponding to the preceding LiDAR readout are fed to the DQN as well. The LiDAR scans are fed to a 1D convolutional neural network (CNN) with the vehicle speeds fed to the last layer where the connections of the DQN are dense. Such a description of the vehicle's state allows the observation of the position, velocity and acceleration of the vehicle in theory. In contrast, the Dreamer model relies on the momentarily LiDAR scan processed with a multi-layer perceptron (MLP) only, heavily depending on Dreamer's ability to recover the partially observed state utilizing the previous observations.

The results of [14] and [17] suggest that either an extended observation space with an advanced neural network architecture or a sophisticated model-based method is needed to realize an RL-based autonomous racing driver. Both options demand on experience with deep neural networks and reinforcement learning, hence, they are suboptimal for a newcomer to the field of autonomous racing.

This paper intends to propose an easily accessible baseline for experimenting with RL in the F1TENTH environment with the following contributions to the field of RL-based autonomous racing.

- A novel LiDAR-only observation is proposed that is expressive regarding vehicle dynamics, including information on velocity and its first and second derivative, acceleration and jerk, respectively.
- The proposed observation is tested with a popular, easily accessible model-free RL method. The trained agent is found to be able to transfer the driving-capability to tracks unseen during the trainings.
- A recommendation is given on selecting the hyperparameters for the training, that is not included in prior work.

The paper is structured as follows. First, the methodology is discussed in Sec. II with a focus on the F1TENTH simulation environment as it needed some minor modifications to be suitable for training an RL agent. The proposed LiDAR-only observation is detailed in Sec. II as well together with the important aspects of the training setup like the reward function and the methodology of defining the hyperparameters. The notable properties of the experiments are described in Sec. II-C with the results introduced presented in Sec. III Finally, a short discussion of the results with an outlook for future work is given in Sec. IV.

II. Methodology

The investigation of the proposed LiDAR-only observation depends both on the simulation of the vehicle dynamics in conjunction with the LiDAR sensor and the training of an RL agent. As modifications and considerations had to be taken in both directions, they are discussed separately.

A. Autonomous vehicle simulation with F1TENTH

The F1TENTH Gym environment [5] is an open-source project which provides realistic vehicle simulation for reinforcement learning tasks. The stepping of the physics simulation is done by the agent, resulting in a deterministic execution, and also enabling faster than real time simulation. The F1TENTH racetrack project proves over 20 racetracks, scaled down for the F1TENTH Gym. Each track includes the centerline, and the track width to the left and right as well.

The project has multiple git branches, our work is based on the main branch. Since we wanted to use Stable-Baselines3 [18], we had to upgrade the OpenAI Gym [19] to the more recent Gymnasium library.

F1TENTH Gym uses a bicycle model to determine the movement of the car. Our work used the following parameters:

TABLE I: Bicycle model parameters

Parameter	Unit	Value
Surface friction coefficient	-	1.0489
Cornering stiffness coefficient, front	1/rad	4.718
Cornering stiffness coefficient, rear	1/rad	5.4562
Distance from center of gravity to front axle	m	0.15875
Distance from center of gravity to rear axle	m	0.17145
Height of center of gravity	m	0.074
Total mass of the vehicle	kg	3.74
Moment of inertial of the entire vehicle about the z axis	kgm2	0.04712
Minimum steering angle constraint	rad	-0.4189
Maximum steering angle constraint	rad	0.4189
Minimum steering velocity constraint	rad/s	-3.2
Maximum steering velocity constraint	rad/s	3.2
Velocity at which the acceleration does not affect wheel spin	m/s	7.319
Maximum longitudinal acceleration	m/s2	9.51
Minimum longitudinal velocity	m/s	-5
Maximum longitudinal velocity	m/s	20
Width of the vehicle	m	0.31
Length of the vehicle	m	0.58

B. Pure LiDAR-based control with model-free reinforcement learning

Model-free and model-based reinforcement learning are two approaches within the broader field of reinforcement learning, each with its own characteristics, advantages, and limitations [20].

In model-free reinforcement learning, the agent learns to make decisions by interacting with the environment without building an explicit model of the environment dynamics, not requiring knowledge of the dynamics or transitions of the environment.

In model-based reinforcement learning, the agent constructs an internal model of the environment to simulate its dynamics. This model is then used for planning and decision-making. The agent uses the learned model to simulate possible future trajectories and then chooses actions that lead to the most favorable outcomes.

Proximal Policy Optimization (PPO) is a model-free reinforcement learning algorithm introduced by Schulman et al. [21]. The term "proximal" in PPO indicates that the optimization is constrained to ensure that the new policy does not deviate too far from the old policy. This constraint helps stabilize the training process and prevents large policy updates that could lead to instability. We selected PPO, because our action and observation space is both continuous, and it is relatively sample efficient compared to other reinforcement learning algorithms.

They hyperparameters were selected based on recommendations from RL Zoo [22], which is a repository containing many already tuned environments. The repository's content is separated into each algorithm available in Stable-Baselines3. We compared our use-case to other environments, and were looking for ones that have similar complexity and are have tuned hyperparameters. We have selected the HalfCheetahBulletEnv-v0 [23] as a base for our parameter set. This environment consists of a 2-dimensional robot, with the goal of moving the robot forward by controlling its joints. We analyzed the parameters and found the following the most impactful for our use-case:

- Enabling State-Depended Exploration [24]
- Changing the activation function to ReLU in both the policy and the value network
- Changing the neural network architecture to contain two hidden layers, both with 256 nodes

For modeling the agent's state we used the last four LiDAR scans. A LiDAR scan consists of 1080 points in the simulator, each point measuring the distance of an obstacle in a given direction. These measurements were normalized by clipping every value above 30.0, and then dividing by 30.0. The rationale behind having four scans is that from two scans the agent can infer its velocity, from three its velocity change, and from four its acceleration change. Since this state description does not have any track specific information, our expectation was that the agent is going to generalize well, being able to navigate tracks it has not seen before, for the cost of more training time.

In formulating the reward structure, we wanted to ensure that it is easy to implement, proven to work, and allow us to compare different agents trained on various tracks. For these reasons the reward function is articulated as the difference between the advancement along the central trajectory at the current time step and the corresponding advancement along the central trajectory in the previous step(1). This approach is similar to the one used by Bosello et al. [17] and was successful with a model-based method.

$$r = progress(t) - progress(t-1) \tag{1}$$

The action space was designed to closely match the F1TENTH Gym simulator's API. The API expects the steering angle in radian, and the desired velocity in m/s. PPO calculates a distribution centered around 0, and as a best practice we normalized the action space to be in the [-1, 1] interval. The output of PPO is then mapped to [-3, 3] in case of the steering angle, the negative values steer the vehicle in the left hand side direction, and the positive values steer in the right hand side direction. In case of velocity, the output of PPO will be scaled to [0, 7.5] interval, 0 meaning the vehicle will come to a halt. 7.5 m/s max speed is realistic in terms of a real F1TENTH car, the racetracks are also scaled down and are usually 2 meters wide.

C. Experiments

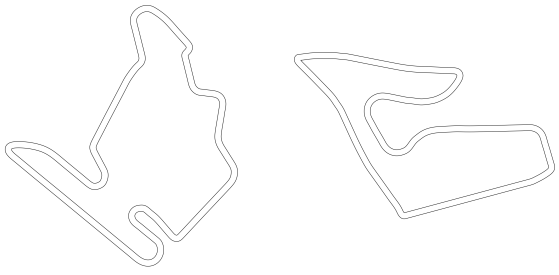
The experiments were set up in order to showcase two properties of the proposed approach of RL-based autonomous racing. On the one hand, the viability of the approach had to be justified. On the other hand, the capability of transferring the knowledge to unseen tracks (or in other words: generalizability) was investigated to find out whether the proposed approach is on par with prior solutions.

Similar to prior works [14], [17], 4 different racetracks were used in the experiments from the collection provided in [25], namely Budapest, Catalunya, Silverstone and Spielberg. The tracks were chosen such that 3 of them overlap with the ones used by Bosello et al. in [17] and were downscaled to match the size of the F1TENTH racecar. The outlines are depicted in Fig. 1.

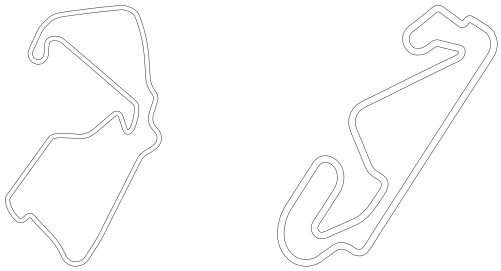
The agent was trained on two of the racetracks: Budapest and Spielberg. Only the resulting model from the Budapest track was evaluated on all of the tracks while the Spielberg model was evaluated only on Spielberg to serve as a reference for the other model.

III. Results and evaluation

The training of the agent was done on a consumer-grade laptop which had a dedicated NVIDIA T550 GPU. We were training on 16 environments in parallel, but due to the laptop's limited processing capabilities these were



(a) Budapest (b) Spielberg



(c) Silverstone (d) Catalunya

Fig. 1: Racetracks used in the study from the collection provided by [25].

_curve-Budapest.jpg _curve-Budapest.bb

_curve-Spielberg.jpg _curve-Spielberg.bb _curve-Spielberg.jpg height

Fig. 2: Training curves on the Budapest (top) and on the Spielberg (bottom) racetracks. The first training step where the agent was able to take a full lap is denoted with a vertical line.

not utilized to the fullest. Training of a model for 10 million steps took about 2.5-3 hours, which allows quick iteration of our approach.

The training curves from the Budapest and the Spielberg track are depicted in Fig. 2 on the top and the bottom, respectively. In both of the cases, the agents were trained for 10M steps and the models were exported regularly and their performance was assessed. The first time when an agent was able to finish a lap is denoted with a vertical line in Fig. 2. As the reward – and in conjunction with it, the agent's performance – fluctuated over the training, the model corresponding to the maximum reward was evaluated instead of the model corresponding to the last training step.

Despite the differences in the achieved maximum reward, both agents were able to take a full lap on its training track in the first quarter of the training. While the length of the original tracks are almost the same[1], the raceline[2] lengths are different. This is reflected in the steps needed to learn to go around the track: the agent achieved the first full lap during training in Spielberg, where the raceline is ≈10% shorter than in Budapest.

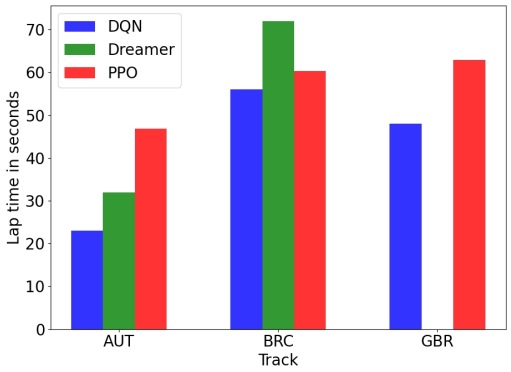


Fig. 3: Laptimes of three different RL algorithms on the Spielberg (AUT), on the Catalunya (BRC) and the Silverstone (GBR) racetrack. DQN is reported in [17], Dreamer is reported in [14] and while PPO is our solution. Please note, that both DQN and Dreamer were trained on the AUT racetrack, while every of the presented tracks were unseen tracks for the PPO during training.

The performance of the agent trained on Budapest was then evaluated on all of the racetracks and is referred to as the PPO model from now on. The laptimes achieved with PPO together with the ones reported with the Dreamer model [14] and the DQN model [17] are depicted in Fig. 3.

While all of the depicted tracks were unseen to the PPO agent during training, both the DQN and the Dreamer agents were trained on the Spielberg (AUT) track. The most notable result is that the PPO agent was able to finish all the – unseen – laps, while Dreamer could not finish on Silverstone. On the Catalunya racetrack, our approach outperformed Dreamer by finishing closely behind the DQN agent. Our model gave the worst performance on the Spielberg track by finishing last with a laptime almost twice as much than needed for the DQN agent.

As the other models were trained on these tracks, the first guess was that if the PPO had been trained on the same track, it would have been performed better. This hypothesis were not supported by the results of the PPO model trained on the Spielberg track as it achieved almost the same laptime when trained on Budapest and evaluated on Spielberg. In fact, the trajectories of the two PPO agent trained on different tracks and evaluated on Spielberg resulted in similar trajectories, depicted in Fig. 4.

A jerky movement is also observed in Fig. 4 that is in

[1]The length of the Budapest and the Spielberg racetrack is 4.4 km and 4.3 km, respectively.

[2]The raceline is provided by Betz et al. in [25] and is calculated by minimizing the summed curvature of the planned trajectory.

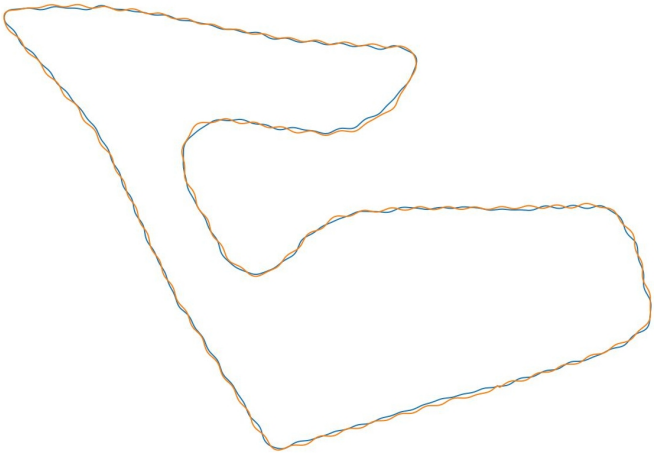


Fig. 4: Vehicle trajectories on the Spielberg racetrack with the agent trained on Budapest (blue) and Spielberg (orange).

line with the findings reported in [14], [17]. Both authors mitigated this effect with punishing the steering actions but Brunnbauer et al. [14] that even the vehicle's motion get smoother, there is no benefit of punishing the steering actions in terms of laptime. As our model performed well in terms of the original goals, we left the investigation of this behavior for a further research.

IV. Discussion

A simple, yet effective approach to realize an RL-based autonomous racing driver is presented in this paper. This work is distinguished from prior with combining a model-free RL technique with a LiDAR-only observation. Both of these aspects are appealing for a newcomer to the scene as the proposed observation fits to the deep neural networks used by the out-of-the-box algorithms of the popular RL-frameworks. On top of that, a methodology for quickly selecting the hyperparameters is also recommended in the paper. As the best practice is to conduct a time-consuming hyperparamter optimization to find out the suitable values, this recommendation can also hold a value for beginners in the field.

Aside of realizing autonomous driving, the proposed method outperforms the existing model-based solution in terms of generalizability and achieves competitive laptimes on some of the unseen racetracks even though the vehicle's trajectory is seemingly jerky. This kind of motion is reported from the literature with other RL-techniques and is found to be cured with the extension of the reward function. Hence, the investigation of punishing the steering action is a direction for developing the presented work.

The most straightforward continuation is however, testing the trained agent's capabilities on a real F1TENTH car by fitting it to a suitable architecture like the one presented in [26]. The presented method is intended not to stand in the way of such a challenge by having simple demands from its environment.

Acknowledgment

The research was supported by the National Research, Development and Innovation Office. (2020-2.1.1-ED-2021-00162)

References

[1] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, D. Hoehener, S.-Y. Liu, M. Novitzky, I. F. Okuyama, J. Pazis, G. Rosman, V. Varricchio, H.-C. Wang, D. Yershov, H. Zhao, M. Benjamin, C. Carr, M. Zuber, S. Karaman, E. Frazzoli, D. Del Vecchio, D. Rus, J. How, J. Leonard, and A. Censi, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, May 2017.

[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA grand challenge," Journal of Field Robotics, vol. 23, no. 9, pp. 661–692, sep 2006.

[3] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth, M. Lienkamp, and B. Lohmann, Indy Autonomous Challenge - Autonomous Race Cars at the Handling Limits. Springer Berlin Heidelberg, 2022, pp. 163–182.

[4] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," in Proceedings of the NeurIPS 2019 Competition and Demonstration Track, ser. Proceedings of Machine Learning Research, H. J. Escalante and R. Hadsell, Eds., vol. 123. PMLR, 08–14 Dec 2020, pp. 77–89. [Online]. Available: https://proceedings.mlr.press/v123/o-kelly20a.html

[5] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "textscf1tenth: An open-source evaluation environment for continuous control and reinforcement learning," in NeurIPS 2019 Competition and Demonstration Track. PMLR, 2020, pp. 77–89.

[6] B. Németh, M. Fazekas, Z. Bagoly, P. Gáspár, and O. Sename, "Lpv-based control design with guarantees: a case study for automated steering of road vehicles," in 2023 European Control Conference (ECC). IEEE, Jun. 2023.

[7] Y. Mizushima, I. Okawa, and K. Nonaka, "Model predictive control for autonomous vehicles with speed profile shaping," IFAC-PapersOnLine, vol. 52, no. 8, pp. 31–36, 2019.

[8] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Durr, "Super-human performance in gran turismo sport using deep reinforcement learning," IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 4257–4264, jul 2021.

[9] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs, L. Gilpin, P. Khandelwal, V. Kompella, H. Lin, P. MacAlpine, D. Oller, T. Seno, C. Sherstan, M. D. Thomure, H. Aghabozorgi, L. Barrett, R. Douglas, D. Whitehead, P. Dürr, P. Stone, M. Spranger, and H. Kitano, "Outracing champion gran turismo drivers with deep reinforcement learning," Nature, vol. 602, no. 7896, pp. 223–228, feb 2022.

[10] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," Nature, vol. 620, no. 7976, pp. 982–987, aug 2023.

[11] B. Evans, H. A. Engelbrecht, and H. W. Jordaan, "Reward signal design for autonomous racing," in 2021 20th International Conference on Advanced Robotics (ICAR). IEEE, dec 2021.

[12] ——, "Learning the subsystem of local planning for autonomous racing," in 2021 20th International Conference on Advanced Robotics (ICAR). IEEE, Dec. 2021.

[13] B. D. Evans, H. W. Jordaan, and H. A. Engelbrecht, "Comparing deep reinforcement learning architectures for autonomous racing," Machine Learning with Applications, vol. 14, p. 100496, Dec. 2023.
[14] A. Brunnbauer, L. Berducci, A. Brandstatter, M. Lechner, R. Hasani, D. Rus, and R. Grosu, "Latent imagination facilitates zero-shot transfer in autonomous racing," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, May 2022.
[15] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," arXiv preprint arXiv:1912.01603, 2019.
[16] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," arXiv preprint arXiv:2010.02193, 2020.
[17] M. Bosello, R. Tse, and G. Pau, "Train in austria, race in montecarlo: Generalized rl for cross-track f1tenth lidar-based races," in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE, Jan. 2022.
[18] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html
[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," arXiv preprint arXiv:1606.01540, Jun. 2016.
[20] R. S. Sutton, Reinforcement learning, 2nd ed., ser. Adaptive computation and machine learning, A. Barto, Ed. Cambridge, Massachusetts: The MIT Press, 2020, literaturverzeichnis: Seiten 481-518.
[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, Jul. 2017.
[22] A. Raffin, "Rl baselines3 zoo," https://github.com/DLR-RM/rl-baselines3-zoo, 2020.
[23] 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, Oct. 2012.
[24] A. Raffin, J. Kober, and F. Stulp, "Smooth exploration for robotic reinforcement learning," in Conference on Robot Learning, 2021.
[25] 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.
[26] Z. Demeter, P. Bogdán, . Bogár-Németh, and G. Bári, "Scalable supervisory architecture for autonomous race cars," in 2024 IEEE Intelligent Vehicles Symposium (IV) (Submitted), 2024.