Digital Twin of a Driver-in-the-Loop Race Car Simulation with Contextual Reinforcement Learning

Siwei Ju [1,2], Peter van Vliet[1], Oleg Arenz[2], Jan Peters[2]

***Abstract*—In order to facilitate rapid prototyping and testing in the advanced motorsport industry, we consider the problem of imitating and outperforming professional race car drivers based on demonstrations collected on a high-fidelity Driver-in-the-Loop (DiL) hardware simulator. We formulate a contextual reinforcement learning problem to learn a human-like and stochastic policy with domain-informed choices for states, actions, and reward functions. To leverage very limited training data and build human-like diverse behavior, we fit a probabilistic model to the expert demonstrations called the *reference distribution*, draw samples out of it, and use them as context for the reinforcement learning agent with context-specific states and rewards. In contrast to the non-human-like stochasticity introduced by Gaussian noise, our method contributes to a more effective exploration, better performance and a policy with human-like variance in evaluation metrics. Compared to previous work using a behavioral cloning agent, which is unable to complete competitive laps robustly, our agent outperforms the professional driver used to collect the demonstrations by around 0.4 seconds per lap on average, which is the first time known to the authors that an autonomous agent has outperformed a top-class professional race driver in a state-of-the-art, high-fidelity simulation. Being robust and sensitive to vehicle setup changes, our agent is able to predict plausible lap time and other performance metrics. Furthermore, unlike traditional lap time calculation methods, our agent indicates not only the gain in performance but also the driveability when faced with modified car balance, facilitating the digital twin of the DiL simulation.**

***Index Terms*—reinforcement learning, imitation leaning, autonomous agent, autonomous racing**

I. INTRODUCTION

SIMULATION has become an indispensable tool in the modern automotive industry. In addition to the various techniques and applications used for vehicle simulation, incorporating a driver model has gained significant traction because it is crucial to consider the impact of drivers during the earliest stages of automotive development. Possible use cases are, for instance, energy consumption prediction [1], user-specific advanced driver assistant systems [2], personalized route planner [3], and hardware-in-the-loop test benches [4].

The motorsport industry, which requires rapid development, places even greater demands on high-quality driver models for realistic lap simulations during extensive setup tests and tuning. Obtaining such models is particularly challenging due to the increased complexity and precision required. To improve performance, the race car and its driver need to be considered as an entity and individual driving styles of professional race drivers need to be considered. As testing on the real car is expensive, only possible in the final development phase and usually regulated and limited, Driver-in-the-Loop (DiL) simulators as shown in Fig. 1 are widely used in top-class motorsport teams [5]. However, DiL simulations are expensive, time consuming, and therefore, limited in terms of the amount of tests that can be executed within a certain time frame. In order to efficiently scale up the number of tests, a full digital twin that includes representative race driver models is highly desirable.

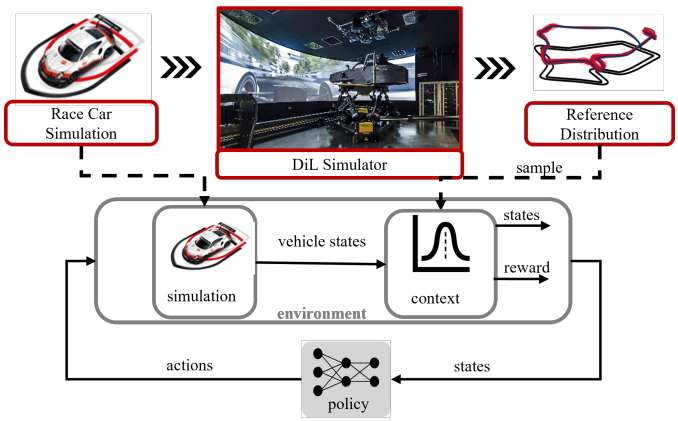

Figure 1: Our framework leverages the vehicle dynamics model used in the DiL simulator to train and evaluate the agent while using the demonstrations generated by professional race drivers to provide the agent with context information during rollouts. These demonstrations are encoded into the *reference distribution*, allowing unlimited samples to be drawn for each rollout, enabling a probabilistic agent with human-like variance.

Developing such driver model for race car driving is especially challenging for the following reasons: 1) the car is being handled on its limits, which requires the driver model to react to rapidly changing and unstable dynamics, 2) a human decision process is difficult to model due to many influencing factors, including experience, multimodal perception, psychological factors, and natural stochasticity, 3) imprecision of human drivers and changing properties of the car and the environment are generating variance in the trajectory of the car, and 4) It is usually expensive to get adequate and high-quality demonstration data.

Research on driver models for urban driving and motorsport scenarios has been ongoing for decades, with various ap-

Manuscript received: November 25, 2022; Revised March 23, 2023; Accepted April 28, 2023.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by Dr. Ing. h.c. F. Porsche AG.

[1]Siwei Ju and Peter van Vliet are with Dr. Ing. h.c. F. Porsche AG, 71287 Weissach, Germany siwei@robot-learning.de, peter.van_vliet@porsche.de

[2]Siwei Ju, Oleg Arenz and Jan Peters are with Intelligent Autonomous System, Computer Science Department, Technische Universität Darmstadt, 64289 Darmstadt, Germany oleg.arenz, jan.peters@tu-darmstadt.de

proaches such as vehicle dynamics analysis [6], control theory [7], machine learning [8], and reinforcement learning [9][10]. However, so far no driver modeling approach was found that addresses all of the challenges and serve as a digital twin for DiL simulations.

Our work aims to address the listed challenges and develop a driver model that can potentially facilitate the digital twin of DiL simulations. Our contribution is formulating race car driving as a contextual reinforcement learning problem, and leveraging human demonstrations as context to enhance performance and efficiency, while learning human-like behavior and variance.

As shown in Fig. 1, we initialize the agent with behavioral cloning, and use reinforcement learning to improve the performance and robustness. Having only limited demonstrations, we fit them into a prior probability distribution called the *reference distribution*, and draw samples from it as the context during explorations. Using *context-aware* states and rewards informed by domain-knowledge, we train the policy by optimizing the expected return over all instances of the references, contributing to a more effective exploration and a stochastic policy with human-like variance, compared to the default step-based Gaussian noise for explorations.

In practice, the demonstrations are generated by professional drivers in the DiL simulator in Fig. 1. During the demonstration and training, the setting of the car is the same as in the professional races, without any driver assistance system such as traction control or anti-lock braking system (ABS), making it more challenging to control.

Being trained and evaluated in the same simulation environment as in the DiL simulator with a simplified track model, our agent outperforms top-class professional race car drivers by 0.4 secs on average on a real race track in comparable scenarios. In addition, being optimized over the reference distribution, the agent learns a stochastic policy, exhibits realistic and human-like variance in terms of the driving line, the speed profile and the lap time. The agent is capable of completing laps even when faced with altered vehicle characteristics such as changes in power, grip, or balance, while predicting plausible lap time changes. The data generated by the agent can be used to predict metrics such as lap time, top speed, and driveability[1].

Related Work

There has been considerable research conducted in various fields that aim to address similar use cases, particularly in the domain of autonomous vehicles and racing.

Lap Time Simulation. Lap time simulation is essential in Motorsport for rapid tests and assessments of vehicle setups without expensive real-track or DiL tests. One of the currently widely used methods is Lap Time Calculation based on Quasi-Steady-State analysis (LTC) [11]. Using the g-g-v[2] diagram, the lap time is estimated by calculating the speed profile based on the assumption of the full utilization of the steady-state potential, with a predefined and discretized driving line. LTC has three main limitations: a) the full dynamic vehicle model is replaced by a simplified, steady-state model, b) consequently, while generating the speed profile, it does not consider the effects that a real driver can take advantage of in real-time, dynamic forward simulation, and c) as the simulation is assuming steady state conditions, it is challenging to assess the notion of driveability.

(Learning-based) Control Theory. The problem of lap time prediction has also been approached by control theory, using optimal control [7] or model predictive control with an analytical or a learned model [12]. Those methods require a vehicle dynamics model, which is usually highly linearized and/or simplified. Furthermore, while these approaches tackle the first two limitations listed before, the third limitation of missing indication of driveability remains unsolved [7].

Reinforcement Learning. Most recently, reinforcement learning (RL) has been adopted for autonomous racing, on model cars [13] or in simulation environments [14][15]. In the video game Gran Turismo Sport, RL outperforms human players [10][16]. The works presented in this field offer valuable insights into the potential of RL. However, they have certain limitations that may hinder their ability to serve as a reliable digital twin for professional race car development. For instance, the simplified vehicle dynamics simulation and the use of driver assistance systems may not accurately capture the complexities and nuances of real race driving conditions. Additionally, the robustness and sensitivity towards vehicle setups are not analyzed, which may lead to suboptimal performance compared to that of professional race drivers. Furthermore, the absence of analysis when the vehicle is pushed to its handling limit can result in non-optimal behavior. It is also important to note that current focus of RL research tends to be on achieving deterministic super-human performance, rather than simulating human-like diverse behavior, which is crucial for understanding the driveability of the car. In summary, while RL has been proved to have great potential in the field of race car simulation, there are still several challenges that must be overcome before it can be used as a reliable digital twin for race car drivers.

Imitation Learning. To take human factors into consideration, different imitation learning techniques such as behavioral cloning ([8], [17]) and generative adversarial imitation learning ([18]) have been used for either urban driving or motorsport scenarios, to learn a policy based on demonstrations. Among them, Probabilistic Modeling of Human Driver Behavior (ProMoD) investigates the potential of behavioral cloning in race car simulation, and it succeeds in learning a stochastic policy with near-to-expert performance. However, its limitations hinder it from being used in practice. First of all, behavioral cloning lacks robustness due to the covariance shift [19]. Adaptation needs to be done through post-processing and is moreover limited in cases that it can handle[20]. Secondly, the performance is worse than human demonstrations. Last but not least, it requires a large amount of demonstration data up to several hundreds of laps which are difficult and expensive to acquire.

[1]Driveability is a notion of a measure for how easy it is to drive a car at the handling limits.

[2]The g-g-v diagram shows the acceleration potential of a car at a given speed.

Despite the research in different fields, there is not yet a method, to the best of our knowledge, which can fulfill the requirements that are necessary to be used as a digital twin for professional race car simulations.

II. METHODOLOGY

Our objective is to learn a driving policy π: $\boldsymbol{a}_t = \pi(\boldsymbol{s}_t)$, where $\boldsymbol{a}_t$ represents the driver inputs given to the car, mapped from the current state represented by the state vector $\boldsymbol{s}_t$. In this section, we first formulate the contextual reinforcement learning problem, followed by the design of the action space and the state space of the policy network. Subsequently, we introduce the *reference distribution* which is fitted from demonstration data collected from the DiL simulator and provides the agent with context information.

A. Problem Formulation

We formulate this problem as a contextual reinforcement learning problem [21]. The latent context is given by reference τ_{ref}, sampled at the beginning of each episode. It remains constant during each episode while affecting the system dynamics, the state space and the reward. The goal is to optimize the policy π with respect to the expected return over all instances of the reference driving lines (τ_{ref}):

$$\max_{\pi} \int_{\tau_{\text{ref}}} q(\tau_{\text{ref}}) \iint_{s_{\tau_{\text{ref}}}, a} \mu^{\pi}(s_{\tau_{\text{ref}}}, a) R_{\tau_{\text{ref}}}(s_{\tau_{\text{ref}}}) \, \mathrm{d}s_{\tau_{\text{ref}}} \, \mathrm{d}a \, \mathrm{d}\tau_{\text{ref}}, \tag{1}$$

where the *reference distribution* $q(\tau_{\text{ref}})$ is the probability density function over trajectories learned from human demonstrations, and $\mu^{\pi}(s_{\tau_{\text{ref}}}, a)$ is the stationary distribution following the policy π [22]. With the states s and the reward function R being context-specific, the agent is able to leverage the context information. Being optimized over the reference distribution, the policy is robust towards different reference lines, learns to optimize its performance based on them, and is able to reproduce the variance presented by the reference distribution.

Reward The reward should reflect the learning objective, which is to optimize the lap time. We did initially test such sparse reward function, giving one reward only at the end of each lap[3]. However, we were unable to learn a reasonable behavior within our computational budget, despite the availability of expert demonstrations. Our initial policy, learned by behavioral cloning, was not even able to make the first turn, and thus the agent received no reward to help it improve its performance. Therefore, we use a context-aware dense reward to encourage better explorations. In addition, by punishing certain unwanted situations, the learning process can be sped up towards an expected driving behavior. The reward function consists of the following terms with corresponding weighting coefficients.

$$R_t = d_{\text{ref}} - c_{\text{off-track}} \mathbb{1}_{\text{off-track}} - c_{\text{slow}} \mathbb{1}_{\text{slow}}, \tag{2}$$

[3]A lap is usually one to two minutes except for some long tracks, and in our simulation, it is sampled at 100 Hz, resulting in a minimal episode length of around seven thousand.

where the performance term d_{ref}, the safety margin $c_{\text{off-track}} \mathbb{1}_{\text{off-track}}$ and speed margin $c_{\text{slow}} \mathbb{1}_{\text{slow}}$ are defined as follows.

- The performance term d_{ref} is defined as the traveled distance along the given reference driving line on the last time step. Another way to understand it is as the velocity component projected onto the direction of the reference line. When optimizing the discounted sum of this term, the policy aims to maximize the average speed, which in turn minimizes the lap time. This reward encourages progress along reference lines instead of the centerline. Cooperating with the context-aware states, the agent learns to use the context information presented by the various references during the explorations, and optimize the performance.
- We punish certain situations, such as off-track or the speed lower than a predefined threshold are $c_{\text{off-track}}$ and c_{slow}, respectively. When the early termination (ET) criteria are triggered, the exploration will be immediately terminated and the corresponding punishment is given. The ET criteria are defined by the indicator functions

$$\mathbb{1}_{\text{off-track/slow}} = \begin{cases} 1 & \text{if off-track / } v_t < v_{\text{threshold}} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

  Off-track means that the agent leaves the track area. The speed threshold $v_{\text{threshold}}$ is a track-dependent parameter. This parameter is chosen small enough to allow for optimal behavior and large enough to help exploration.

In our experiments, the coefficients $c_{\text{off-track}} = 10$ and $c_{\text{slow}} = 100$ are tuned to achieve a certain compromise between performance and safety.

Based on the reward, we find that the agent needs to look far ahead, for instance, to plan for the coming straight line after one corner already from the corner entry. This can be achieved by using a large discount coefficient (γ in Table I).

B. Policy

We parameterize the policy π using a feed-forward neural network with state vector $\boldsymbol{s}_t$ and actions $\boldsymbol{a}_t$ as follows:

Action. The action $\mathbf{a}_t$ consists of a combined acceleration signal α and the steering wheel angle δ: $\boldsymbol{a}_t = [\alpha_t, \ \delta_t]^T$, where the acceleration signal is combined from acceleration and brake, α_t = accelerator pedal actuation - brake pedal actuation. Combining both signals is motivated by the fact that the driver usually doesn't brake and accelerate simultaneously, and it speeds up the exploration significantly, as those two signals are closely related, and the transition from one to the other is usually continuous.

States. The decision making process of human drivers is based on many influencing factors, including the sense of balance from a combination of visual, audio, and haptic perception, prior knowledge gathered from previous laps, and their intentions. Since we cannot simulate the multi-sensor perception, it is critical to provide the agent with informative features. Based on those considerations, informed by domain-

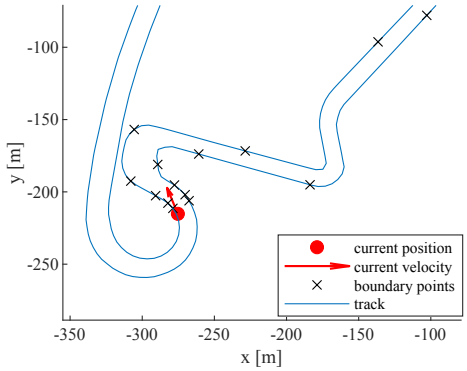

Figure 2: Boundary points on a real race track. The x,y coordinates of the points on the track borderlines are calculated in the current coordinate framework of the car with the velocity as the positive x direction and positive y direction pointing to the left.

knowledge and modified by experiments, we present a feature set consisting of the following states:

$$\boldsymbol{s}_t = [v, a_x, a_y, \beta_F, \beta_R, r_F, r_R, \boldsymbol{\alpha}_{\text{position}}, \boldsymbol{c}_{\text{poly}}, \alpha_{\text{offset}}, d_{\text{offset}}],$$

which can be divided into three types:

- Vehicle states: absolute velocity (v), longitudinal acceleration (a_x), lateral acceleration (a_y), average front/rear slip ratio (r_F, r_R), average front/rear slip angle (β_F, β_R)[4]. Compared with other existing works ([8], [10]), this part is significantly expanded. Those signals contribute to the sense of balance and capture of the vehicle dynamics, which in reality comes from multi-sensor perception of human beings.
- Position features: previous work uses rangefinder features [10] to provide information about the track. However, we propose the boundary point features as shown in Fig. 2. It contains the relative position of a series of points on the track borders, with distance markers relative to the current position of the center of gravity of the car [5, 10, 20, 40, 80, 160, 320, 640] on both sides of the track boundaries. We combine both visual perception on the track in the short range and the prior knowledge of track layouts in the long range in this single set of features, which can be easily acquired in our setting, without time-consuming computations.
- Path planning features: A polynomial fitting of a local path $\boldsymbol{c}_{\text{poly}}$, and the angular and distance offset ($\alpha_{\text{offset}}, d_{\text{offset}}$) in a predefined preview time [8]. They are computed based on the reference driving line used for each roll-out. These features contain prior knowledge and experiences of the driver embedded in the reference trajectory, and the short-term path planning information depending on the current states.

Using these states, the agent is able to get information about the car, its position on the track, and its relative position to the reference to be context-aware.

[4]Standard definitions as in [23] are adopted here.

C. Context: Reference Distribution

The context is given by the reference driving line and reflected in the states and the reward function so that the agent can take advantage of them during explorations, improve the efficiency and performance, and in the end, show corresponding variance in behavior.

The most straight-forward choice for the reference driving lines is to use the human demonstrations. However, it has multiple limitations: a) it is expensive to get adequate demonstration data for enough context instances, b) due to the natural non-deterministic behavior of human, distractions during simulator sessions and the limited amount of laps, the raw demonstration data are usually sparsely distributed and include outlier laps that are less representative.

In order to make use of limited expert data efficiently and get rid of the influence of outliers, we first fit the demonstration data into a probabilistic distribution called *reference distribution* using Probabilistic Movement Primitives (ProMP) [24]. To get a compact but informative representation, the spatial information in the form of the x and y coordinates of the driving line is projected into a low-dimension weight space of a series of equally distributed radial basis functions with the phase variable of track distance s. For each single trajectory in the demonstrations, transformed from originally time-based to distance-based, the weight coefficients $\boldsymbol{w}$ of the basis functions are fitted using ridge regression. Then, we derive the *reference distribution* by fitting weight vectors of all N demonstrations to a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$, by $\boldsymbol{\mu}_{\boldsymbol{w}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{w}_i$ and $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{w}_i - \boldsymbol{\mu}_{\boldsymbol{w}})(\boldsymbol{w}_i - \boldsymbol{\mu}_{\boldsymbol{w}})^T$.

Now, the distribution of driving lines on each track is described efficiently with this *reference distribution*, and subsequently, by sampling weight vectors $\boldsymbol{w}^* \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ and transforming back to the distance space, an arbitrary number of driving lines which are similar to the demonstrations can be generated. Those samples are subsequently used as the reference driving lines to present the agent with the context for each roll-out during explorations and test runs.

III. EXPERIMENTS AND EVALUATION

We train our agent based on a demonstration data set with six laps[5] from one professional race driver on a real race track as shown in Fig. 4. For reinforcement learning, the implementation of Proximal Policy Optimization (PPO) [25] from stable-baselines [26] is adopted, with default hyperparameters apart from the ones listed in Table I. We disabled advantage normalization for a consistent reward tuning.

For intellectual property protection, all plots are shown normalized or in relative value, omitting the unit.

Method evaluation Firstly, we investigate the impact of context information and context awareness on the training process and performance. Fig. 3 displays the average discounted reward across training epochs for three experiments that share the same settings, except for variations listed below. Additionally, we include the results from the best behavioral cloning experiment that we have conducted.

[5]Each lap is around 70 seconds and sampled in 100 Hz, resulting in a demonstration data set of around 40,000 data points.

Table I: Parameter and Configuration of the Experiment

component	parameter	value
reference distribution (ProMP)	sigma of basis functions	0.002
	regularization factor	1^{-15}
	num basis function	700
PPO	num step	2048
	num env	16
	batch size	64
	gamma γ	0.998
	lam	0.98

Table II: Lap Time Performance - Experiment Settings

experiment	A	B	C
best lap time	1.0000	1.0103 (+ 0.705 sec)	1.0320 (+2.212 sec)

- A: baseline setting on track 1, 2, and 3. The reward and the states are context-aware.
- B: centerline reward. With the reward term d projected on the track center line, the reward is not context-aware.
- C: no context. States relevant to the reference driving line are removed and the states are not context-aware.
- D: behavioral cloning (BC). The agent is trained using behavioral cloning and the return is calculated using reward in Setting A for the rollouts between the training epochs until it starts to overfit. [6]

The rewards are calculated based on their own reward-references to investigate the impact on the speed of convergence. To analyze their performances, the best lap time performance of each setting is listed in Table II in relative to baseline setting A. Their driving lines and actions are in Fig. 4 and Fig. 5 correspondingly, except for BC which is not able to complete the laps under the same condition in our experiments.

We summarize our main findings as follows:

- The introduction of context in the reward and states (A) leads to more efficient learning and significantly better lap times by enabling the agent to explore effectively, compared to only PPO-default Gaussian noise (C).
- Using context-reward (A) instead of centerline reward (B) enables the agent to better use various reference trajectories, resulting in more effective exploration and slightly better final lap time performance.
- The context provided by the references, which are similar to human demonstrations, contributes to the agent's ability to imitate the driving line and actions. The agent's ability to imitate the demonstrations improves as it relies more on the context information. (Utilization of context: A (context in states and reward) $>$ B (context in only states) $>$ C (no context), imitation of the driving line and the actions: (A $>$ B $>>$ C)

Performance In addition to comparing different experiment settings, we evaluate the performance of the agent against that of a human driver. The driving lines of one agent lap and one human lap, both with average lap time, are plotted in Fig. 4. The corresponding actions, the speed profile, and the time difference are plotted in Fig. 5. Our agent gains the greatest advantage on T2/T3, where there is a successive direction changing corners (a chicane) which is challenging to handle. From the driving line plot on Fig. 4 it can be clearly seen that the model uses the entire track to get the best performance, similar to professional human driving styles. It is also worth noting that the agent learns to apply also the very subtle brake and accelerator pedal applications that the human driver executes, at every part of the track, for instance at track progress 0.15 and 0.55 on Fig.5.

[6]We have conducted experiments for behavioral cloning with different feature sets, neural network structures, hyperparameters, and amounts of data. For clarity, we show here the best result that we have observed, using the setting from [27]. We have concluded that significantly more demonstrations are necessary for the BC agent to pass the first turn, and around 120 laps on three tracks are used for the result shown in Fig. 3.

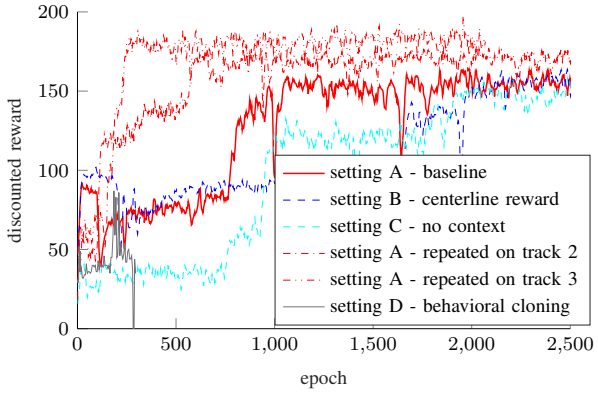

Figure 3: Learning curves of different experiment settings A, B, and C, with the baseline setting A on additional track 2 and 3 to demonstrate the transferability of the method.

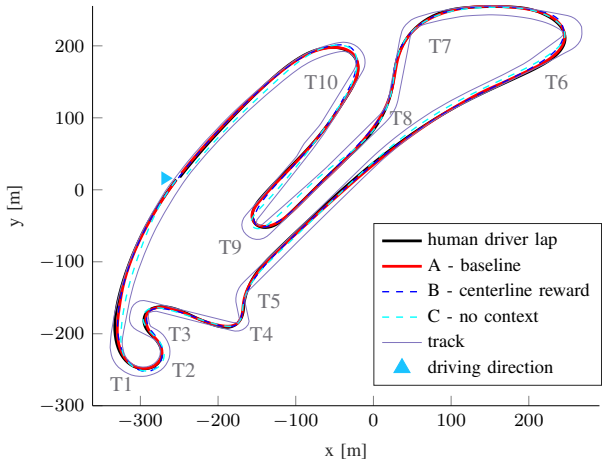

Figure 4: The driving lines are plotted for one human driver lap with the average lap time, and the average lap of the agents trained with the three settings.

Transferability Being trained on a single track, our agent often struggles to complete laps when deployed on new tracks. However, we observe that retraining the agent leads to a significant improvement in performance. As demonstrated in Figure 3, our experiments with the baseline setting A on two additional tracks show that the agent converges efficiently to competitive performance after being retrained.

Distribution and variance Apart from the competitive performance, as the agent is optimized over the reference distribution, it is able to generate realistic and human-like

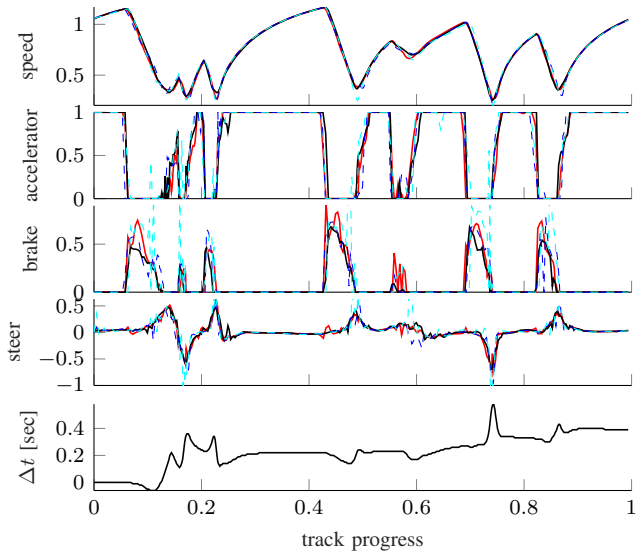

Figure 5: Detailed comparison of laps. The speed profile, the actions (in normalized values) for the four laps are shown in corresponding to Fig. 4. In addition, the profile of Δt, which is the time difference of the two laps when the track progress on the x-axis is reached, between the human driver lap and the baseline agent lap (setting A) is plotted in black. A positive value means that the agent lap in red is faster.

Table III: Lap Time Performance - Agent vs Human

	fastest	average
human	0.9990	1.0000
agent	0.9932	0.9941

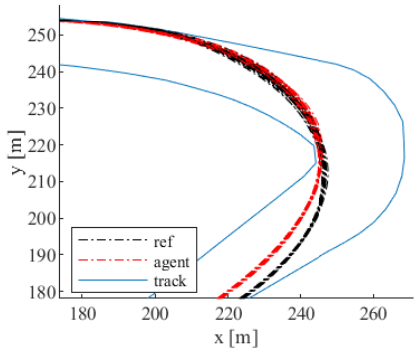

Figure 6: Driving line distribution zoomed in for T6. The track is shown by the blue track borderlines. Twenty reference driving lines and twenty agent laps are plotted in black and red, respectively. The laps are counterclockwise.

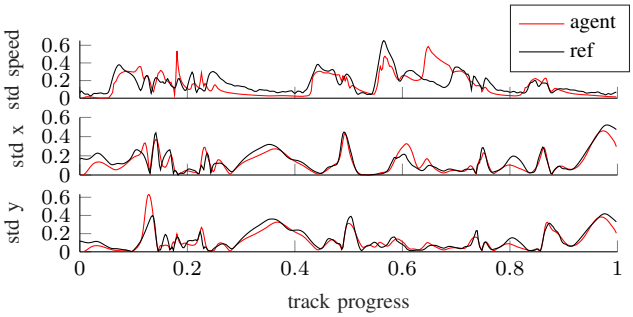

Figure 7: Standard deviation of the speed profile and the driving line along the track, calculated based on the twenty reference and agent laps shown in Fig. 8.

variance in driving line, speed profile and lap time, using different references. It achieves better performance compared to the human demonstrations in terms of the fastest and the average lap time as listed in in Table III. The numbers are shown as a ratio relative to the average lap time of the human driver. The driving lines driven by the agent have comparable diversity to the references as shown in in Fig. 6. Globally analyzed, the ability of the agent to reproduce the variance presented in the reference distribution is elaborated by Fig. 7 by the standard deviation along the track. The agent is able to generate similar level of variance on both the speed profile and the driving line throughout the entire lap, while significantly optimizing the lap time, as shown in Fig. 8. Since only driving lines are provided as the context, the agent learns to get the best performance out of each reference, and presents a lap time distribution with less variance as it optimizes the lap time performance. In contrast, the laps generated by the agent under Gaussian noise show noisy signals and the lap times are around 0.2 percent worse.

Robustness and setup sensitivity To be used for setup analysis as a digital twin of the DiL simulator, the agent needs to be robust towards setup changes in power, global grip and the car balance which we can modify by, for instance, changing front lateral grip. We train the agent on a given baseline setup and subsequently test it by directly deploying it on tuned car setups without retraining. The slip angles and slip ratios in the states contributes to the robustness because the agent learns the limit of the car through explorations, avoids pushing it over the limit, and completes the laps given that the setup is tuned within a predefined range where the limit of the car does not change dramatically.

In Fig. 9, the result of power sensitivity analysis is taken as an example and shown in detail, compared with LTC results. The agent succeeds in staying within a small range around the LTC line, meaning that it is able to use the full potential of a given setup, even if it deviates from the setup that it is trained on. Since the power influences mainly the acceleration behavior of the car, the agent only needs to brake with the correct combination of brake point and speed, using the information provided by the boundary points feature. This brake behavior is consistent with what it learns during

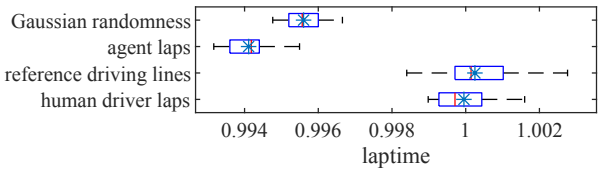

Figure 8: Distributions of lap times shown in box-plot for the demonstration data set with six laps, 20 reference trajectories, 20 agent laps following the references, and 20 agent laps with the Gaussian randomness in actions. The top and bottom edges of the box indicate the 25th and 75th percentiles.

Table IV: Sensitivity Analysis

setup item	sensitivity	agent	LTC
lap time	power	$-4.47 \cdot 10^{-4}$	$-4.67 \cdot 10^{-4}$
	global grip	$-3.54 \cdot 10^{-3}$	$-3.76 \cdot 10^{-3}$
	longitudinal grip	$-1.41 \cdot 10^{-3}$	$-1.87 \cdot 10^{-3}$

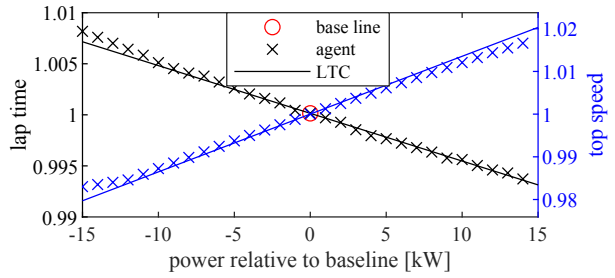

Figure 9: Power sensitivity analysis. The red circle shows the baseline, on which the agent is trained. The agent is then tested on cars with power in the range of [-15 kW, 15 kW]. It manages to complete the laps in all the tests, and the resulting lap time and top speed are shown with 'x' in black and blue, respectively. Sensitivities based on LTC are shown with solid lines in the same color. The LTC results and the agent laps are normalized with their lap times with respect to the baseline (relative power = 0 kw).

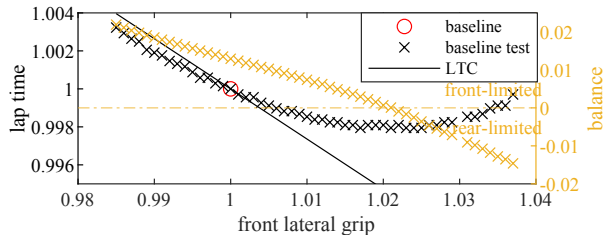

Figure 10: Front lateral grip analysis. The agent, trained on the baseline setup, is represented by the red circle and its roll-outs on varying front lateral grip level are plotted with the black 'x's. The balance metric in yellow is calculated by taking the difference between the front and the rear slip angle. A positive value indicates that the performance is limited by the available grip on the front wheels.

exploration. Furthermore, the lap time sensitivity and top speed sensitivity for global grip, longitudinal grip, and lateral grip, aggregated in Table IV. The values result from dividing the normalized lap time difference by the setup parameter change. They are very similar to the sensitivities calculated by the steady-state lap time calculations (LTC)[11]. In all setup sensitivity analysis that we have done, the agent shows reliable robustness and reasonable sensitivities.

Driveability and adaptation In addition to the well known and rather straightforward sensitivity analysis shown before, the balance[7] of the car and how it influences the performance is more complicated to investigate.

Our agent indicates not only the performance but also the driveability of certain setups. We evaluate the sensitivity of the agent to the front lateral tire grip, which can be directly tuned in simulation and strongly influences the car balance, as depicted in Fig. 10. The roll-outs of the agent present a 'U' shape in terms of the lap time, with the optimum around 1.02, while LTC predicts linear sensitivity. On the left hand side of the graph, where the car is more limited by the front axle, the lap time decreases with the increasing of the front lateral grip. While the front lateral grip keeps increasing and the car starts to get limited by the rear axle, as indicated by the balance metric, the agent starts to make mistakes, struggling with oversteers, and the lap time cannot be improved further.

Same tendency is also shown in Fig. 11 by the lap distributions. The trained agent runs 50 roll-outs with different reference samples on each front lateral grip value to both sides until no laps can be completed and we plot the lap time distribution of the completed laps. Then we are able to use linear regression to identify three stages. During stage I and III, the number of completed laps experiences a significant drop with a similar slope to the left and the right. This could be attributed to the combination of decreased driveability and the agent operating out-of-distribution. But on stage II, to the left of the baseline setup, the number of completed laps increases and throughout stage II, the number of completed laps decreases, with an improvement in lap time across the distribution. But there is a clear increase in variance.

The observations, especially during stage II, indicate that that increasing the front lateral grip may decrease the driveability of the setup, while potentially improving lap time performance, if it is tuned within a range around the baseline.

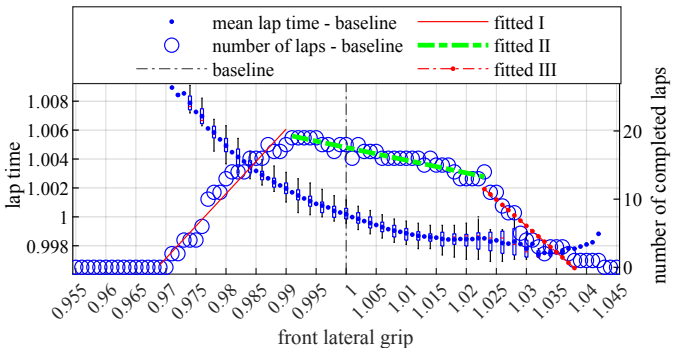

Figure 11: Driveability analysis of shifted car balance. The lap time distributions of the trained baseline agent with 50 samples of reference driving lines on cars with tuned front lateral grip in a range 0.955 to 1.045, symmetric around the baseline setup 1.0, are plotted, using the same annotation as in Fig. 8. Using linear regression, the number of completed laps is fitted into three stages.

IV. CONCLUSION

In this paper, we develop a contextual reinforcement learning framework to train a race car driving policy for professional high-fidelity race car simulation. By fitting demonstration data collected from the human driver-in-the-loop simulator into a probabilistic model, our approach requires only a few demonstrations. Samples from the reference distribution are used as reference driving lines during exploration, and provide context information to the reinforcement learning agent. We investigate different actions, states and reward functions based on domain-knowledge and experiments. This agent is robust and competitive with realistic variance as present in the demonstrations and outperforms professional race car drivers in terms of the fastest lap time. It shows not only human-like diverse driving lines and speed profiles, and also the ability to predict reasonable lap time and driveability when faced with setup changes. Therefore, it can be potentially used as a digital twin of the DiL simulator, for carrying out large

[7]The balance indicates if the car is front-limited (it has more grip on the front tires than the rear), neutral, or rear-limited. The balance can be tuned by varying the lateral grip of the front tires. With more grip available on the front tires, the car is less front-limited and more difficult to handle. However, talented and well-trained drivers are able to gain better performance with it up to a certain amount.

numbers of high-quality tests, facilitating rapid development in the motorsport industry.

For future work, we would like to integrate a more realistic track model into the simulation, so that details such as curbs or local grip can be taken into consideration. Methodologically, developing a track- and setup-universal agent that can operate on various tracks and setups and reach the optimums without retrain is a compelling idea. However, to achieve this goal, it necessitates additional setup and position features and training on multiple tracks. Furthermore, more research on imitating individual driving styles of human drivers is of great interest to provide human-like driveability predictions.

ACKNOWLEDGMENT

The authors gratefully acknowledge the computing time provided to them on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt. This is funded by the Federal Ministry of Education and Research, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities.

Furthermore, we would like to thank Kai Fritzsche for his support on setting up the high-fidelity race car dynamics model for training and Theo Ackerman for the investigation on the position features.

REFERENCES

[1] T. Wilhelem, H. Okuda, B. Levedahl, and T. Suzuki, "Energy consumption evaluation based on a personalized driver-vehicle model," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2016.

[2] J. M. Fleming, C. K. Allison, X. Yan, R. Lot, and N. A. Stanton, "Adaptive driver modelling in ADAS to improve user acceptance: A study using naturalistic data," *Safety Science*, vol. 119, pp. 76–83, nov 2019.

[3] S. Funke and S. Storandt, "Personalized route planning in road networks," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, nov 2015.

[4] W. Liu, Q. Song, Y. Li, and W. Zhao, "A novel driver model for real-time simulation on electric powertrain test bench," in *SAE Technical Paper Series*. SAE International, oct 2017.

[5] K. Fritzsche, F. Strecker, and M. Huneke, "Simulation-based development process for motorsport and series production," *ATZ worldwide*, vol. 119, no. 9, pp. 60–63, 2017.

[6] B. Siegler, "Lap time simulation for racing car design," Ph.D. dissertation, University of Leeds, 2002.

[7] N. Dal Bianco, E. Bertolazzi, F. Biral, and M. Massaro, "Comparison of direct and indirect methods for minimum lap time optimal control problems," *Vehicle System Dynamics*, vol. 57, no. 5, pp. 665–696, 2019.

[8] S. A. Löckel, "Machine learning for modeling and analyzing of race car drivers," 2022.

[9] J. C. Kegelman, "Learning from professional race car drivers to make automated vehicles safer," Ph.D. dissertation, Stanford University, 2018.

[10] 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. [Online]. Available: https://doi.org/10.1109%2Flra.2021.3064284

[11] D. Brayshaw, "Use of numerical optimisation to determine on-limit handling behaviour of race cars," Ph.D. dissertation, Cranfield University, 2004.

[12] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, oct 2019.

[13] 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.

[14] E. Perot, M. Jaritz, M. Toromanoff, and R. de Charette, "End-to-end driving in a realistic racing game with deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[15] K. Güçkıran and B. Bolat, "Autonomous car racing in simulation environment using deep reinforcement learning," in *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2019, pp. 1–6.

[16] 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.

[17] H. Wei, W. Ross, S. Varisco, P. Krief, and S. Ferrari, "Modeling of human driver behavior via receding horizon and artificial neural network controllers," in *Conference on Decision and Control*. IEEE, 2013, pp. 6778–6785.

[18] A. Kuefler and M. J. Kochenderfer, "Burn-in demonstrations for multi-modal imitation learning," in *International Conference on Autonomous Agents and Multi-Agent Systems*. IFAAMAS, 2018, pp. 1071–1078.

[19] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 661–668. [Online]. Available: https://proceedings.mlr.press/v9/ross10a.html

[20] S. Löckel, S. Ju, M. Schaller, P. van Vliet, and J. Peters, "An adaptive human driver model for realistic race car simulations," Mar. 2022.

[21] A. Hallak, D. D. Castro, and S. Mannor, "Contextual markov decision processes," Feb. 2015.

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 2017.

[23] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*. Society of Automotive Engineers Warrendale, 1995.

[24] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[26] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.

[27] S. Löckel, A. Kretschi, P. van Vliet, and J. Peters, "Identification and modelling of race driving styles," *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–29, 2021. [Online]. Available: https://doi.org/10.1080/00423114.2021.1930070