This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2024.3423630

Active Exploration in Iterative Gaussian Process Regression for Uncertainty Modeling in Autonomous Racing

Tommaso Benciolini[1], Chen Tang[2], Marion Leibold[1], Catherine Weaver[3], Masayoshi Tomizuka[3], Wei Zhan[3]

***Abstract*—Autonomous racing creates challenging control problems, but Model Predictive Control (MPC) has made promising steps toward solving both the minimum lap-time problem and head-to-head racing. Yet, accurate models of the system are necessary for model-based control, including models of vehicle dynamics and opponent behavior. Both dynamics model error and opponent behavior can be modeled with Gaussian Process (GP) regression. GP models can be updated iteratively from data collected using the controller, but the strength of the GP model depends on the diversity of the training data. We propose a novel active exploration mechanism for iterative GP regression that purposefully collects additional data at regions of higher uncertainty in the GP model. In the exploration, a MPC collects diverse data by balancing the racing objectives and the exploration criterion; then the GP is re-trained. The process is repeated iteratively; in later iterations, the exploration is deactivated, and only the racing objectives are optimized. Thus, the MPC can achieve better performance by leveraging the improved GP model. We validate our approach in the highly realistic racing simulation platform Gran Turismo Sport of Sony Interactive Entertainment Inc for a minimum lap time challenge, and in numerical simulation of head-to-head. Our active exploration mechanism yields a significant improvement in the GP prediction accuracy compared to previous approaches and, thus, an improved racing performance.**

***Index Terms*—Autonomous racing, trajectory planning and tracking, interaction, learning for control, active exploration, Gaussian Processes**

I. Introduction

AMONG the many applications of autonomous driving, autonomous racing has recently gained increased attention in research [1], also for real-world tests like Roborace and the Indy Autonomous Challenge. Two scenarios are considered: the minimum time trial and head-to-head racing against an opponent. In the former, a single race car drives around a constrained track trying to minimize the lap time. In this scenario, the main control challenges arise from pushing the vehicle to the handling limits, a task that expert humans can do well, but is challenging for control algorithms. In particular, physics-only models typically used in urban or highway environments are not well suited to represent the vehicle dynamics close to the handling limits. Aerodynamics forces, nonlinear deformations, nonlinear tire dynamics, and weight transfer in the vehicle caused by accelerating, braking, or steering are effects typically neglected when deriving a simplified nominal vehicle dynamics model [2] for computationally efficient online control. Yet, they must not be ignored if the goal is to push the vehicle to its handling limits and minimize the lap time. Within model-based controllers, Model Predictive Control (MPC) relies on a prediction model of the vehicle, and the input is determined by iteratively solving an optimal control problem over a finite horizon. Thus, the large uncertainty introduced by the modeling errors when the vehicle is driven near handling limits must be accounted for, for example adding a learning component to the physics-based model [3]–[5]. Other works have addressed the challenge considering Bayesian approaches [6], [7]. Furthermore, it has been shown that model-free reinforcement learning can outperform human performance [8]. However, a main challenge with learning-based methods is to obtain data sufficiently representative, while still avoiding dangerous situations.

In the scenario with an opponent, the controlled vehicle, named Ego Vehicle (EV) in this work, must compete with another agent and perform overtaking maneuvers. In this case, the uncertainty about vehicle dynamics near the handling limits is compounded by another major challenge: the interaction with the other agent, which is a well-researched problem for autonomous urban and highway driving [9], [10]. Both enforcing collision avoidance and planning a successful overtaking maneuver require the EV to handle the uncertainty around the unknown future position of the opponent. Initial approaches considered *passive* prediction models, that is, predicting the future trajectory of other agents given historical data and the current traffic configuration. Such approaches allow for a simplified planning framework, in which the future trajectories of other agents are assumed to be independent of the current decision of the EV. However, in highly interactive scenarios, such as automated racing, where the other agent is a competing opponent, the reaction of other agents to the EV decision must be considered. Knowledge of the opponent's reaction to its own future trajectory is crucial to allow safe and efficient overtaking maneuvers. To account for the reaction to own decisions, the opponent can be represented as a rational agent in a game-theoretic framework [11], which is, however, computationally demanding. Alternatively, the policy of the

Manuscript received Month Day, Year; revised Month Day, Year.

[1]T. Benciolini and M. Leibold are with the Chair of Automatic Control Engineering at the Technical University of Munich, Germany (email: {t.benciolini;marion.leibold}@tum.de). This work was developed during T. Benciolini's visit to the University of California, Berkeley.

[2]C. Tang is with the Department of Computer Science at the University of Texas at Austin, TX, USA (email: chen.tang@utexas.edu). Work done when C. Tang was with the University of California, Berkeley.

[3]C. Weaver, M. Tomizuka, and W. Zhan are with the Department of Mechanical Engineering at the University of California, Berkeley, CA, USA (email: {catherine22;tomizuka;wzhan}@berkeley.edu).

opponent to the current and past configurations and the EV own decision can be learned from data [12], [13]. In the latter case, it is fundamental to retrieve a training dataset sufficiently representative to allow for reliable learning of the policy.

In this work, we address both sources of uncertainty in autonomous racing, i.e., modeling errors in the dynamics and the representation of the unknown policy of the opponent accounting for the reaction to the EV's own decisions, using an iterative Gaussian Process (GP) regression algorithm, following the approach from [14]. GP regression is a non-parametric machine learning framework that provides uncertainty measures over its prediction based on previously collected measurements. Our main contribution is extending the iterative GP framework by adding an active exploration mechanism designed to retrieve representative data and improve learning performance. Previous works in autonomous racing did not consider the active exploration of the feature space to improve the learning performance and relied on data collected while maximizing the EV performance. Compared to other learning tools, such as artificial neural networks, a major advantage of GPs is that a measure of the model uncertainty is provided, which we exploit in the exploration mechanism to yield an enriched dataset and, ultimately, an improved prediction. In our algorithm, the dataset of the GP is updated iteratively with the measurements collected over several runs, re-training the model when the dataset is updated. During the initial iterations, the reference trajectory of the EV is designed to encourage the exploration of the regions of the feature space with a high posterior covariance of the prediction error. In doing so, the dataset is rapidly replaced with properly selected data points that refine the learning performance.

We find that enriching the dataset through the active exploration mechanism yields a significant improvement in the learning performance and, eventually, in the EV performance. We show that the GP exploration algorithm can be applied successfully both when the GP model is used for error compensation in the minimum lap time task and for the opponent modeling in head-to-head racing. For the minimum lap time, we test the algorithm in the highly realistic racing simulation platform Gran Turismo Sport of Sony Interactive Entertainment Inc [15], where high-fidelity dynamics models are used to simulate the vehicle, and we offer a comparison with the previous work [14]. For the task with the opponent, we compare our algorithm with the previous work [13] in the simulation environment therein provided.

The contributions of this work are as follows:

- We propose an iterative GP regression framework with an active exploration mechanism which uses heuristics to explore the most uncertain regions of the state space as indicated by the GP posterior covariance matrix;
- We implement the active exploration mechanism in the objective of the model predictive controllers for both a time trial race and a head-to-head racing challenge;
- We show that our method, which combines uncertainty-based exploration with training dataset selection of the most diverse data, improves the GP prediction accuracy and the EV racing performance in comparison simulations with previous approaches from the literature.

In Section I-A, we review the relevant related work, whereas Section II-A and II-B present preliminaries regarding the vehicle model and GP regression, respectively. Our novel iterative GP regression framework with active exploration is presented in Section III-A for the time trial and Section III-B for the opponent challenge. The validation simulations for both autonomous racing scenarios are presented and analyzed in Section IV. In Section V we discuss the limitations of the current approach and possible improvements. The conclusion with an outlook for future research is given in Section VI.

A. Related Work

In this section, we give a concise review of the relevant related work concerning GP regression for the compensation of vehicle dynamics and opponent modeling in autonomous racing, as well as existing approaches for active exploration presented in other fields. Further relevant works on automated racing can be found in the recent survey [1].

Vehicle Dynamics and GP Compensation Models: Various physics-based vehicle models have been proposed depending on what assumptions are valid for the application [2], [16]. The dynamic bicycle model is common in control algorithms and models the dynamics of a single-track vehicle with two wheels [2]. The lateral tire forces may be assumed to be linear with respect to the slip angle of the tires [2], which is valid for the low slip angles encountered during autonomous urban and highway driving. However, racing vehicles operate in the non-linear, saturated regions of the tire dynamics, and phenomena such as drifting and weight transfer have a significant effect on planning and control [17]. While parameterized tire models like Pacejka's Magic Formula [18] are more descriptive, it can be challenging to identify all of the parameters of the Magic Formula. Furthermore, modeling errors may persist due to weight transfer, suspension dynamics, or the lumped tire dynamics of the bicycle model. Therefore, rather than spending significant engineering efforts attempting to model every detailed aspect of the vehicle dynamics, learning-based approaches could leverage system data to improve model accuracy and control performance in a more efficient way.

Recent efforts have explored how Gaussian Processes (GP) can compensate for modeling errors in real-time control [19], [20]. The GP adds to a nominal vehicle model and is trained to improve the model's accuracy on data collected from the actual system [14]. GPs, unlike standard neural networks, provide an estimate of the posterior covariance that can be used to predict model uncertainty during control [3]. Retraining the GP after collecting more data can increase model accuracy and ultimately improve the performance of planning or control [14], [20]. However, data collected during normal EV operation might not be sufficiently representative of the EV dynamics in all situations. Thus, the GP compensation model can be further improved if trained on a more diverse dataset. For example, in vehicle racing, acceleration limits can be progressively updated from collected data to safely expand safe operation as the vehicle improves [21]. Furthermore, active exploration specifically targets underrepresented regions to collect additional data and reduce the GP's model uncertainty.

Opponent Modeling: The autonomous racing scenario with an opponent has been considered in [11] using a game-theoretic framework, in which the policy of the EV is chosen as a Nash equilibrium, following well-established approaches for urban and highway autonomous driving [22]–[25]. However, the solution of a dynamic game is generally a computationally expensive task. Moreover, accurate knowledge of the opponent's own reward function and constraints is required to implement this approach, which is limiting in practice. Alternatively, machine learning methods have been used to directly learn the policy or the closed-loop trajectory of the opponent from data. In [12], a GP is used to learn a mapping from the current EV and opponent state to the future opponent state and the posterior covariance of the GP is used to tighten safety collision avoidance constraints. The approach is interesting and relatively computationally inexpensive at run time, however implements a passive interaction approach, in which the reaction of the opponent to the current EV's own decisions is not considered. In [26], in the context of urban autonomous driving, a neural network is used to approximate the closed-loop behavior of other agents in a game-theoretic fashion. Instead of solving an optimization problem to predict the future trajectory of other agents, their reaction to the EV's own decision is predicted by a neural network that takes as input the future state of the EV. However, a neural network does not provide a measure of the uncertainty around the prediction. In [13], a GP is trained in a similar fashion, conditioning on the future plan of the EV as well. However, the model is trained on a dataset of measurements collected during normal operation in several previous runs. As a result, the GP prediction of the opponent is not accurate for all possible overtaking strategies that the EV can attempt. To improve the prediction accuracy, an active exploration mechanism is needed, explicitly targeting more regions of the feature space.

Active Exploration: For learning-based approaches, the choice of the training set plays a major role in determining the performance of the learned model and in the generalization capability. In particular, in iterative approaches, in which the data used are measurements collected during the previous iterations while maximizing the performance of the EV, the dataset might not be sufficiently expressive to significantly improve the model in the whole feature space.

Active learning [27] has been widely investigated in many fields, such as coverage control [28] and autonomous navigation [29]. Recent work has dedicated attention to active learning approaches based on Koopman operators [30] and Bayesian Optimization [31], whose most recent advances are discussed in a recent survey [32]. Further approaches to active learning are also mentioned in the surveys on learning-based MPC [33], active learning in robotics [34], and deep active learning [35]. In particular, approaches for active exploration in GP regression have been proposed for control of wind farms [36], airborne wind energy system [37], [38], or UAV delivery control [39]. In such applications, an accurate and updated estimate of the wind field is fundamental, therefore the referenced works proposed approaches to trade-off between maximizing the performance of the system and controlling it in a way to collect measurements to improve the wind field estimation. In autonomous racing, however, compromising the performance objectives is only acceptable during the early stages of the competition, whereas eventually, the focus must be the maximization of the EV performance. Therefore, the trade-off between exploration and performance objectives must be tuned dynamically. Furthermore, the decision on the regions to be explored must take place in real-time.

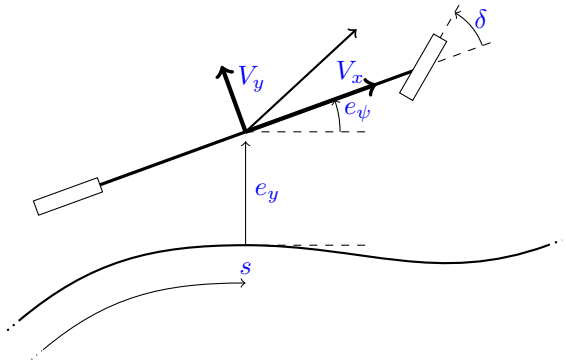


Fig. 1: Scheme of the Dynamic Bicycle model.

II. Preliminaries

In this section, we detail preliminaries for our work. Section II-A describes a dynamic bicycle model that will serve as a nominal vehicle model. Section II-B describes Gaussian Process (GP) models, which are used in later sections for error compensation and opponent modeling.

A. Vehicle Dynamics

We model the racing vehicle dynamics using a dynamic bicycle model [16] referred to the road-aligned Frenet coordinates, as represented in Figure 1. The state of the vehicle is $\boldsymbol{\xi} = [V_x, V_y, \dot{\psi}, e_\psi, e_y, s]^\top$, where V_x and V_y are the vehicle's longitudinal and lateral velocity, respectively, in the vehicle's body frame, $\dot{\psi}$ is the yaw angular velocity, e_ψ and e_y are the yaw angle and lateral displacement of the center of gravity of the vehicle with respect to the reference path, and s represents the traveled distance along the reference path. The relative yaw angle e_ψ and lateral distance e_y in Frenet coordinates are defined with respect to the closest point of the reference path. The input $\boldsymbol{u} = [\delta, a_x]^\top$ consists of the front steering angle and of the longitudinal acceleration resulting from the powertrain, which is applied to the rear wheel. Although simplified, the bicycle model represents a good trade-off between keeping the number of parameters low, which is crucial for real-time computations, and having sufficiently accurate dynamics, that reflect the main characteristics of the motion. Moreover, since a learning-based term is used to compensate for the nominal dynamics, a more complex dynamics model, such as the four-wheel model, might yield only marginal improvements [17].

The dynamics is derived from force-mass and inertia-moment balance, for the first two components V_x and V_y, and then from the kinematics for the other state quantities. The dynamics is

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} a_x - \dfrac{F_{y\mathrm{f}}\sin(\delta) + R_x + F_{x\mathrm{w}}}{m} - g\sin(\varphi) + \dot{\psi}V_y \\ \dfrac{F_{y\mathrm{f}}\cos(\delta) + F_{y\mathrm{r}}}{m} - \dot{\psi}V_x \\ \dfrac{l_{\mathrm{f}}F_{y\mathrm{f}}\cos(\delta) - l_{\mathrm{r}}F_{y\mathrm{r}}}{I_{zz}} \\ \dot{\psi} - \dfrac{V_x\cos(e_\psi) - V_y\sin(e_\psi)}{1-\kappa(s)e_y}\kappa(s) \\ V_x\sin(e_\psi) + V_y\cos(e_\psi) \\ \dfrac{V_x\cos(e_\psi) - V_y\sin(e_\psi)}{1-\kappa(s)e_y} \end{bmatrix}, \tag{1}$$

where m is the mass of the vehicle, I_{zz} is the moment of inertia, and l_{f} and l_{r} represent the distance of the center of gravity from the front and rear axle, respectively. R_x is the tire rolling resistance, and $F_{x\mathrm{w}}$ is the wind drag force applied on the vehicle body. $F_{y\mathrm{f}}$ and $F_{y\mathrm{r}}$ are the lateral tire forces of the front and rear tires, which are nonlinear and vary as the tire slips along the road surface. Furthermore, gravity is acting on the vehicle with acceleration g and φ is the inclination of the road. $\kappa(s)$ is the curvature of the reference path at position s. A more thorough discussion of the model is reported in [17]. In the following, we assume to have access to all state components.

In this work, we employ a discretized version of the model obtained via forward Euler:

$$\boldsymbol{\xi}^+ = \boldsymbol{\xi} + \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})T, \tag{2}$$

where T is the sampling time and $\boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})$ is a compact representation of (1). However, the bicycle model in (1) can be insufficient to reliably describe the vehicle motion at its handling limits, in particular, neglecting the influence of nonlinear deformations, aerodynamics forces, and weight transfer in the vehicle caused by accelerating, braking, or steering. In our previous work [17], it was shown that modeling such effects can improve racing performance. Yet, the racing performance is still worse than the human-best performance. It indicates that purely physics-based models are insufficient, and a data-driven compensation term is necessary for achieving the fastest lap time possible.

It is worth noting that we could potentially adopt a more complicated dynamics model, such as a four-wheeled model, as the nominal dynamics model. However, we found in our previous work [17] that the four-wheeled model only brought a marginal improvement in racing performance compared to the bicycle model and thus introduced unnecessary computational costs to the optimal control problem. Also, introducing a more sophisticated nominal dynamics model, especially one with a higher dimensional state space, results in a larger parameter space for the GP compensation model, since a larger list of features would be necessary to span the feature space. It would then lead to increasing computation time and memory usage to run the active exploration algorithm, which will be apparent after we introduce the algorithm in Sec. III. Furthermore, more data would be required as a result of the enlarged feature space, which means a time-consuming exploration phase and increased training complexity.

B. Gaussian Processes

GP regression is a machine learning method used to infer the value of an unknown function given a dataset of M measurements $\mathcal{D} = \{\boldsymbol{z}_i, \boldsymbol{y}_i\}_{i=1}^{M}$, with $\boldsymbol{z}_i \in \mathbb{R}^{n_{\mathrm{z}}}$ the input features and $\boldsymbol{y}_i \in \mathbb{R}^{n_{\mathrm{y}}}$ the output features. A GP is defined as a collection of random variables, each subset of which is jointly normally distributed, and is fully specified by the prior mean and the kernel used as prior covariance [40]. It is assumed that the underlying unknown function $\boldsymbol{g}(\cdot)$ relates the input and the output features as follows

$$\boldsymbol{y}_i = \boldsymbol{g}(\boldsymbol{z}_i) + \boldsymbol{w}_i, \tag{3}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{n_{\mathrm{y}}}, \boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{w}})$ is i.i.d. Gaussian noise with diagonal covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{w}} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{n_{\mathrm{y}}}^2)$. The unknown function is specified through its mean, which we assume zero without loss of generality, and a kernel function $k^a(\boldsymbol{z}, \boldsymbol{z}')$, where $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_{\mathrm{z}}}$ are two input GP input feature vectors. The scalar function $k^a(\boldsymbol{z}, \boldsymbol{z}')$ is chosen to encode the prior assumptions and the function properties. To approximate the modeling error in the dynamics, we use the squared exponential kernel [40]

$$k^a(\boldsymbol{z}, \boldsymbol{z}') = \sigma_{k^a}^2 \exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{z}')^\top \boldsymbol{L}_{k^a}^{-2}(\boldsymbol{z}-\boldsymbol{z}')\right), \tag{4}$$

with parameter L_{k^a} defining the characteristic length-scale and $\sigma_{k^a}^2$ the squared signal variance, whereas to infer the future trajectory of the opponent, we employ the Matérn kernel with parameter $\nu = 1.5$ [40]

$$k^a(\boldsymbol{z}, \boldsymbol{z}') = \left(1 + \frac{\sqrt{3}\|\boldsymbol{z}-\boldsymbol{z}'\|^2}{l_{k^a}}\right)\exp\left(-\frac{\sqrt{3}\|\boldsymbol{z}-\boldsymbol{z}'\|^2}{l_{k^a}}\right), \tag{5}$$

where l_{k^a} is a length scale parameter. Both kernels are widely used and have been chosen consistently with [14] and [13], respectively, to allow for a comparison in which the effect of our active exploration mechanism can be thoroughly discussed. The parameters of the kernels are optimized by maximizing the marginal likelihood of the observations [40].

The posterior mean and covariance of d-th entry $g_d(\boldsymbol{z})$ of the underlying unknown function $\boldsymbol{g}(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{z}), \boldsymbol{\Sigma}(\boldsymbol{z}))$ at the arbitrary point $\boldsymbol{z}^*$ conditioned on the training set $\mathcal{D}$ are obtained as

$$\mu_d(\boldsymbol{z}^*) = (\boldsymbol{k}^a)^\top \boldsymbol{K}^{-1}\boldsymbol{\gamma}_d \tag{6a}$$

$$\sigma_d(\boldsymbol{z}^*) = k^{a*} - (\boldsymbol{k}^a)^\top \boldsymbol{K}^{-1}\boldsymbol{k}^a, \tag{6b}$$

where $\boldsymbol{k}^a = [k^a(\boldsymbol{z}_1, \boldsymbol{z}^*), \ldots, k^a(\boldsymbol{z}_M, \boldsymbol{z}^*)]^\top$, the entries of matrix $\boldsymbol{K}$ are $K_{ij} = k^a(\boldsymbol{z}_i, \boldsymbol{z}_j)$, $k^{a*} = k^a(\boldsymbol{z}^*, \boldsymbol{z}^*)$, and $\boldsymbol{\gamma}_d = [y_{1,d}, \ldots, y_{M,d}]^\top$ contains the training outputs corresponding to the d-th entry.

III. Method

In this section, we introduce our active exploration framework that can iteratively train GP models in autonomous racing scenarios. The exploration mechanism is augmented into the

objective function of a model predictive controller (MPC), and it strategically explores locations of the state space where the GP model has the highest uncertainty. We first detail the active exploration framework for the time trial racing challenge, in Section III-A, where a GP model is used to compensate for EV modeling error in an offline optimal trajectory planner and online trajectory-tracking MPC. Then, in Section III-B, we consider head-to-head racing with an opponent, where the GP model is used to predict the behavior of the opponent and the EV employs an online MPC for trajectory planning and control. We discuss the necessary changes we made to adapt the proposed active exploration framework to this scenario.

A. Minimum Lap Time Application

In the minimum lap time task, our iterative exploration-based controller is used to compensate for modeling errors in the dynamics when the vehicle approaches handling limits. To minimize the lap time, first, a time-optimal trajectory for the EV is planned, then, the vehicle is driven around the track by the MPC. Relying on the nominal model of the vehicle dynamics does not suffice to minimize the lap time. Therefore, we use a GP compensation model to improve the prediction of the EV state. It is important to account for the influence of unmodeled effects of the dynamics also on the optimal path, therefore we use the GP compensation both in the planning and in the MPC tracking phase, adopting the double GP compensation scheme that was presented in [14]. At the end of the trial, the measurements collected are used to retrain both GP models, and the trials are repeated iteratively.

Our method leverages the uncertainty in the GP prediction as a heuristic to guide active data collection in regions of high uncertainty, with the goal of improving the prediction accuracy of the GP. In the first couple of trials, active exploration takes place; namely, the EV control is determined as a trade-off between the MPC performance objective and the exploration objective. In doing so, the enriched data can be used iteratively to re-train the GP; thus, the GP will be more accurate, particularly in regions of high uncertainty in previous iterations. Our goal is to improve the overall performance of the MPC after exploration is complete, and the MPC may fully exploit the more accurate GP model. In the following parts, we discuss the components and main aspects of our proposal, which is implemented in the online tracking phase for the time trial. The details of the derivation of the time-optimal reference are given in the Appendix for completeness.

GP Model: In general, the GP model predicts an unknown value, $\boldsymbol{y}$, as a function of the current state $\boldsymbol{z}$ according to an unknown function (3). In the case of the time trial, we employ a GP to compensate for the unmodeled dynamics of the vehicle [14]. To reduce the dimensionality of the GP, rather than conditioning the GP to be a function of the full state $\boldsymbol{\xi}$ of the EV, we can condition on an input feature vector, $\boldsymbol{z}$, that is a deterministic, known linear function of the state and of the input of the EV, i.e. $\boldsymbol{z} = \boldsymbol{G}[\boldsymbol{\xi}^\top, \boldsymbol{u}^\top]^\top$, where $\boldsymbol{G}$ is a matrix selecting the relevant features from the EV state $\boldsymbol{\xi}$ and input $\boldsymbol{u}$. $\boldsymbol{G}$ must be designed so that, given $\boldsymbol{z}$, it is possible to reconstruct the original state and input components used in the mapping. With abuse of notation, we denote this operation with $\boldsymbol{\xi}, \boldsymbol{u} = \boldsymbol{G}^{-1}\boldsymbol{z}$.

Here the predicted value is $\boldsymbol{y}_k^{\text{MPC}} = \boldsymbol{\xi}_{k+1} - \boldsymbol{\xi}_{k+1}^{\text{pred}}$, where $\boldsymbol{\xi}_{k+1}$ is the next state and $\boldsymbol{\xi}_{k+1}^{\text{pred}} = \boldsymbol{\xi}_k + \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k)T$ is the next state predicted by the nominal vehicle model from the current state (2). Specifically, if the GP compensation term added to the nominal system dynamics is defined as

$$\boldsymbol{y}^{\text{MPC}} = \boldsymbol{g}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}}) \sim \mathcal{N}\left(\boldsymbol{\mu}^{\text{MPC}}(\boldsymbol{z}^{\text{MPC}}), \boldsymbol{\Sigma}^{\text{MPC}}(\boldsymbol{z}^{\text{MPC}})\right), \tag{7}$$

then the system dynamics can be modeled as

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}_k\boldsymbol{\xi}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{d}_k + \boldsymbol{\mu}^{\text{MPC}}(\boldsymbol{z}^{\text{MPC}}), \tag{8}$$

where μ^{MPC} models the error of the linearized dynamic bicycle model (2) with respect to the real dynamics at the GP input feature $\boldsymbol{z}_{\text{MPC}}$. The mean and standard deviation of the distribution, $\boldsymbol{\mu}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}})$, and $\boldsymbol{\Sigma}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}})$, are obtained from (6) to predict the output $\boldsymbol{y}_i$ from the input feature vector $\boldsymbol{z}_i$. Once the GP model is trained, $\boldsymbol{y}_k^{\text{MPC}}$ can be added to $\boldsymbol{\xi}_{k+1}^{\text{pred}}$ to compensate for modeling errors. For the offline planning problem, the discretized nonlinear dynamics (2) is used and the GP compensation is analogously defined in terms of a residual term.

Following [14], we use GP to compensate the states having the greatest impact on the prediction error, V_y and $\dot{\psi}$, i.e., $\boldsymbol{\mu}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}}) = [0, \mu_{V_y}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}}), \mu_{\dot{\psi}}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}}), 0, 0, 0]^\top$. In this application, the GP input feature vector is $\boldsymbol{z}_{\text{MPC}} = \boldsymbol{G}[\boldsymbol{\xi}^\top, \boldsymbol{u}^\top]^\top = [V_y, \dot{\psi}, \delta]^\top$. Furthermore, defining the $\boldsymbol{z}^{\text{MPC}}$ with respect to a nominal predicted trajectory rather than on the actual predicted state $\boldsymbol{\xi}_k$ and predicted input $\boldsymbol{u}_k$ allows real-time computation of the MPC [14], [17]. Precisely, $\boldsymbol{z}^{\text{MPC}}$ is computed from the nominal state $\tilde{\boldsymbol{\xi}}_k$ and $\tilde{\boldsymbol{u}}_k$ of the linearized dynamics from the previous MPC iteration. We record the values of $(\boldsymbol{z}_k^{\text{MPC}}, \boldsymbol{y}_k^{\text{MPC}})$ during real-time control to construct the training dataset $\mathcal{D}$.

Measurements collected while tracking the optimal path are not necessarily diverse enough to train the GP models since the states encountered will be concentrated in a small subset around the reference path being tracked. It limits the learning performance and, consequently, the improvement yielded by the GP compensation both in the planning and MPC tracking. Therefore, we propose our active exploration scheme aimed at enriching the dataset of measurements.

MPC: The optimal control problem of tracking MPC is

$$\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \sum_{k=0}^{N-1} \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^{\text{ref}}\|_{\boldsymbol{Q}} + r_\delta \Delta\delta_k^2 \tag{9a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{A}_k\boldsymbol{\xi}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{d}_k + \boldsymbol{\mu}^{\text{MPC}}(\boldsymbol{z}^{\text{MPC}}(\tilde{\boldsymbol{\xi}}_k, \tilde{\boldsymbol{u}}_k)), \quad \forall k = 0, \dots, N-1 \tag{9b}$$

$$w_{\text{r},k} + \gamma_k \le e_{y,k} \le w_{\text{l},k} - \gamma_k, \qquad \forall k = 1, \dots, N \tag{9c}$$

$$\boldsymbol{u}_{\min,k} \le \boldsymbol{u}_k \le \boldsymbol{u}_{\max,k}, \qquad \forall k = 0, \dots, N-1, \tag{9d}$$

where N is the MPC prediction horizon. The cost function (9a) is designed to penalize rapid changes in the steering angle according to weight $r_\delta > 0$, with $\Delta\delta_k = \delta_k - \delta_{k-1}$ and δ_{-1} is set equal to the last applied steering angle δ_{t-1} at time $t-1$. $\boldsymbol{Q} \ge 0$ is the weight to penalize deviations of the state

$\boldsymbol{\xi}_k$ from the reference $\boldsymbol{\xi}_k^{\text{ref}}$, which plays an important role in encouraging the exploration of the feature space depending on the value of α. Constraint (9b) relies on a linearized version of the bicycle model dynamics, computed with respect to a nominal trajectory $\tilde{\boldsymbol{\xi}}, \tilde{\boldsymbol{u}}$ [17]. Since the GP model is not embedded into the optimization [17], minimization problem (9) is a quadratic problem that can be solved in real-time.

Our prior work [14] has shown that iteratively collecting data with the MPC and retraining the GP model (7) can improve the performance of the GP prediction and MPC. We further propose a mechanism that will employ the knowledge of the fact that $\boldsymbol{y}^{\text{MPC}}$ is predicted by a GP to purposefully explore regions of the state space where the prediction of $\boldsymbol{y}^{\text{MPC}}$ has larger uncertainty. Inspired by [41], we use large posterior covariance of the GP to indicate regions of the state space that need further exploration. Inspired by [39], we realize active exploration by appropriately changing the state reference $\boldsymbol{\xi}^{\text{ref}}$ in the optimal control problem (9). By doing so, the MPC cost function in (9a) is a function of the predicted states and control actions *and* the posterior covariance of the GP model. The cost function will trade-off between the MPC's original performance objective and the exploration objective as described in the next paragraph.

Active Exploration: The goal of the active exploration mechanism is to solve (9) such that it encourages the exploration of the feature space and collects new measurements that enrich the dataset $\mathcal{D}$. For this purpose, we change the reference $\boldsymbol{\xi}^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}}$, that would be obtained from the optimal planner, to visit states where the uncertainty in the prediction is large.

We consider the feature vector $\boldsymbol{z}$ used for GP prediction and determine its target value $\boldsymbol{z}^{\text{ref}}$ that the MPC should explore to improve the prediction accuracy of the GP, while considering the performance objectives at the same time. The target GP input feature vector $\boldsymbol{z}^{\text{ref}}$ is chosen from a list of n_{G} candidate feature vectors, $\{\boldsymbol{z}^{(i)}\}_i^{n_{\text{G}}}$. The list is designed to span the feature space: First, the candidate values of each feature component are selected to cover its maximum range estimated from collected data; Then, candidate feature vectors are created combinatorially from the candidate values of all the feature components. It is worth noting that by first determining the exploration objective $\boldsymbol{z}^{\text{ref}}$ and following it in terms of tracking a reference trajectory in the MPC cost function, the computational complexity required to solve the optimal control problem can be kept low, in contrast to directly incorporating the feature uncertainty as an exploration intrinsic in the cost function (9a), which will then embed the GP model into the optimization problem and cause non-convexity [36].

Algorithm 1 details the procedure to set a new reference, which takes as input the reference state and action trajectory computed by the offline planner, $\boldsymbol{\xi}_k^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}}$, and outputs the updated reference $\boldsymbol{\xi}_k^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}}$, selected so that the target GP input feature vector $\boldsymbol{z}^{\text{ref}}$ is visited. First, $\tilde{\boldsymbol{z}} = \boldsymbol{G}[(\boldsymbol{\xi}_k^{\text{ref}})^\top, (\boldsymbol{u}_k^{\text{ref}})^\top]^\top$ is computed, that is, the feature vector that would be visited following the trajectory of the optimal planner. Then, the candidate feature vectors $\boldsymbol{z}^{(i)}$ are ranked based on two competing criteria: 1) *their proximity to $\tilde{\boldsymbol{z}}$, the features of the optimal reference state* (lines 3-5). For each candidate feature vector $\boldsymbol{z}_i, i = 1, \ldots, n_{\text{G}}$, its proximity is quantified by the rank of its distance to $\tilde{\boldsymbol{z}}$ among all the candidates, denoted by D_i, with $D_i = 1$ implying the farthest from $\tilde{\boldsymbol{z}}$ and $D_i = n_{\text{G}}$ implying the closest; and 2) *their posterior covariance* (lines 6-8), which are calculated using the GP model and weighted by the matrix S. Analogous to D_i, we introduce the ordering index V_i with the covariance as the sorting criterion, with $V_i = 1$ implying the lowest posterior covariance and $V_i = n_{\text{G}}$ implying the highest posterior covariance. To trade-off between these objectives, we select the target feature vector by maximizing their convex combination with weight $\alpha \in [0, 1]$ (line 9). Once the target feature vector is selected, we find its corresponding state and action to update the reference state used for the tracking MPC (line 10).

Algorithm 1 Active exploration via state selection for the MPC reference

1: **Input**: $\boldsymbol{\xi}_k^{\text{ref}}$, u_k^{ref}, $\{\boldsymbol{z}^{(i)}\}_i^{n_{\text{G}}}$, $\boldsymbol{\Sigma}^{\boldsymbol{\xi}}(\cdot)$ $\boldsymbol{S}$, α
2: $\tilde{\boldsymbol{z}} \leftarrow \boldsymbol{G}[(\boldsymbol{\xi}_k^{\text{ref}})^\top, (\boldsymbol{u}_k^{\text{ref}})^\top]^\top$ # GP feature from racing objectives
Sort candidate feature vectors by their distance to the reference state's feature vector:
3: $d_i \leftarrow \|\boldsymbol{z}^{(i)} - \tilde{\boldsymbol{z}}\|$ $\forall i = 1, \ldots, n_{\text{G}}$
4: $\boldsymbol{d} \leftarrow \text{sort}(d_1, \ldots, d_{n_{\text{G}}})$
5: $D_i \leftarrow i : \boldsymbol{d}[i] = d_i$ $\forall i = 1, \ldots, n_{\text{G}}$
Sort candidate features by their weighted posterior covariance:
6: $v_i \leftarrow \|\boldsymbol{\Sigma}^{\boldsymbol{\xi}}(\boldsymbol{z}^{(i)})\|_{\boldsymbol{S}}^2$ $\forall i = 1, \ldots, n_{\text{G}}$
7: $\boldsymbol{v} \leftarrow \text{sort}(v_1, \ldots, v_{n_{\text{G}}})$
8: $V_i \leftarrow i : \boldsymbol{v}[i] = v_i$ $\forall i = 1, \ldots, n_{\text{G}}$
Select reference features using trade-off parameter α:
9: $\boldsymbol{z}^{\text{ref}} = \boldsymbol{z}^{(i^*)}$, where $i^* \leftarrow \arg\max_i (\alpha V_i + (1 - \alpha) D_i)$
Update the reference with the states in $\boldsymbol{z}^{\text{ref}}$:
10: $\boldsymbol{\xi}_k^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}} \leftarrow \boldsymbol{G}^{-1} \boldsymbol{z}^{\text{ref}}$
11: **Output**: *Updated MPC reference,* $\boldsymbol{\xi}_k^{\text{ref}}, u_k^{\text{ref}}$

The covariance of GP has previously been used as a mechanism for data selection [41], as well as for autonomous racing [19]. By selecting $\boldsymbol{z}^{\text{ref}}$ to balance the distance from the racing objectives and covariance criterion, the MPC balances exploration while remaining near the original MPC reference. Increasing α places more weight on the exploration of the feature space, and when $\alpha = 0$, the MPC defaults to use its standard reference. The target value $\boldsymbol{z}^{\text{ref}}$ should not be too far $\tilde{\boldsymbol{z}}$ (small d_i) to prevent significant deterioration in the performance of the controlled system, and to prevent possibly dangerous behaviors and loss of stability during the exploration. Conversely, $\boldsymbol{z}^{\text{ref}}$ should correspond to values with large posterior covariance, v_i, to explore uncertain regions of the state space. We opt to weight the posterior covariance $\boldsymbol{\Sigma}^{\boldsymbol{\xi}}(\boldsymbol{z}^{(i)})$ by a positive semi-definite matrix $\boldsymbol{S} \geq 0$. The weighing matrix $\boldsymbol{S}$ is a hyperparameter used to reflect the relative importance of the uncertainty in different GP components. We set $\boldsymbol{S}$ equal to the weights from the MPC cost function (9a), with a view to giving priority to features whose posterior covariance is larger for those components that are more relevant for the MPC tracking. It is also worth observing that the posterior covariance for each candidate feature in the list can be computed immediately after the training of the GP and stored prior to MPC run time, significantly reducing the computational demand of Algorithm 1 at run time.

In the first few iterations, we set α to be large, so that the focus is placed on the exploration of the feature space and

the collection of data points that enrich the dataset. In later iterations, α is decreased to zero; thus, the focus is entirely on the performance objectives of the MPC, taking advantage of the accurate GP prediction model obtained from training with the dataset from the exploration.

Remark 1: We adapt Algorithm 1 from [39], but replace the mutual information-based exploration metric used in [39] with the weighted posterior covariance. Computing the mutual information-based exploration metric requires selecting a set of data points to quantify the information gain of a given feature vector candidate. The selection procedure itself introduces additional computational costs. Moreover, computing mutual information can also be time-consuming, since it involves high-dimensional matrix inversion and multiplication. To this end, we use the weighted posterior covariance, which can be directly obtained from the GP model, to simplify the computation. Meanwhile, as we will introduce later, we also use the weighted posterior covariance as a heuristic to diversify the selected training data. We intend to use the same metrics to specify consistent data priority for the active exploration and data selection phases.

***Diverse Data Selection**:* During the repeated trials, a large number of data points are collected. Using all such data to train the GP model is impractical and unnecessary, as a smaller dataset of appropriately selected data points suffices to represent the input-output relation. However, creating a smaller dataset by randomly sampling from the collected data points, as in [14], does not guarantee that the diverse data points collected during the exploration phase are appropriately exploited. For this reason, at the end of each iteration, we train the GP using a smaller dataset of points obtained with the data selection approach described in [41], outlined in the following.

The goal is to select a (small) collection of points $\mathcal{D} = \{(\boldsymbol{z}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ to represent the feature space and allow GP to predict as accurate as possible. When adding a datum point $(\boldsymbol{z}_i, \boldsymbol{y}_i)$ to the dataset or replacing existing ones, our policy leverages a similarity measure between the new datum point and the present collection, namely, the posterior prediction covariance (6b) at $\boldsymbol{z}_i$ given all other data points in the dataset $\mathcal{D}$. The policy to update the dataset works as follows:

- If a datum point's posterior covariance given the current dataset is larger than the median of the posterior covariance of all data points currently in the dataset, for at least one of its output features, it is added to the dataset;
- If the dataset is full, the new datum point replaces the data point in the dataset with the smallest posterior covariance.
- If the data points yielding the lowest posterior covariance for different output dimensions are different, we consider the dimension in which the ratio between the new posterior covariance of the new point and the minimum posterior covariance of points in the dataset is the largest. Moreover, we use the outlier rejection mechanism described in [19, Section V-B].

In contrast to [19], we do not consider a decay factor to encourage the removal of older data points first, with the goal of prioritizing data points that contribute the most to maximizing the data covariance, rather than the most recent points. Older data points that have been collected during the exploration in previous iterations are, in general, more significant than recent points collected during the last iterations, in which the focus is on the maximization of the performance.

Remark 2: Two GP models are used to compensate for the errors in the dynamics, one during the planning phase and one during the MPC tracking phase. The two GP models differ since they compensate for different nominal dynamics, namely the nonlinear dynamics used in the offline planning problem and the linearized dynamics used in the MPC tracking problem, respectively. Consequently, two different datasets are extracted separately from the set of data collected during the trials, with each dataset diversified considering the prediction error and covariance with respect to the nominal dynamics used in the planning and MPC tracking phase respectively.

***Constraint Tightening**:* Constraints (9c) and (9d) ensure that the lateral position of the vehicle and the input stay within the track bounds and the actuation bounds, respectively. Since the prediction of the lateral error in (9b) is influenced by the GP compensation $\boldsymbol{g}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}})$, in contrast to [14], we tighten the constraints to address the uncertainty in the prediction. Taking uncertainty into account in the constraints is crucially important to reduce the risk of dangerous movements of the EV during the exploration phase. The modeling error $\epsilon_{y,k}$ for e_y at prediction step k is an affine transformation of Gaussian variables, therefore, is also Gaussian distributed. Thus, the support of the uncertainty $\epsilon_{y,k}$ is unbounded, and a robust tightening, guaranteeing constraint satisfaction for all realizations of the uncertainty $\epsilon_{y,k}$, is not possible. Hence, we implement a stochastic tightening requiring

$$\Pr\left[e_{y,k} + \epsilon_{y,k} \leq w_{\text{l},k}\right] \geq \beta, \tag{10}$$

where $0 \leq \beta \leq 1$ is the risk parameter. Constraint (10) yields a deterministic formulation for the tightening parameter γ_k in (9c). The covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{\xi}}$ of the predicted state $\boldsymbol{\xi}_k$ at step $k = 1, \ldots, N$ is obtained recursively from the dynamics (9b) and from the covariance $\boldsymbol{\Sigma}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}})$ of the GP compensation $\boldsymbol{g}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}})$ as

$$\boldsymbol{\Sigma}_{k+1}^{\boldsymbol{\xi}} = \boldsymbol{A}_k \boldsymbol{\Sigma}_k^{\boldsymbol{\xi}} \boldsymbol{A}_k^{\top} + \boldsymbol{\Sigma}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}}), \tag{11}$$

where for all prediction steps k the state $\boldsymbol{\xi}_k$ and the GP compensation $\boldsymbol{g}^{\text{MPC}}(\boldsymbol{z}_{\text{MPC}})$ are uncorrelated because the compensation is computed from a nominal trajectory, $\tilde{\boldsymbol{\xi}}_k, \tilde{\boldsymbol{u}}_k$. From $\boldsymbol{\Sigma}_k^{\boldsymbol{\xi}}$, the covariance $\sigma_{e_y,k}^2$ of the prediction error $\epsilon_{y,k}$ at prediction step k is obtained and the tightening parameter γ_k is computed as in [42]

$$\gamma_k = \sqrt{2}\sigma_{e_y,k}\text{erf}(2\beta - 1). \tag{12}$$

Because of symmetry, the same tightening parameter is applied to the lower bound in (9c). In the presence of large uncertainty and high probability β, parameter γ_k might grow up to the point that the set of feasible positions for the EV is prohibitively small or even empty. This is an inevitable consequence of the unbounded support of the (Gaussian) uncertainty distribution. To prevent this eventuality, a saturation mechanism on γ_k can be further introduced.

B. Head-to-Head Racing Application

In head-to-head racing, the EV needs to predict the opponent's future trajectory in order to plan overtaking maneuvers. The opponent's future trajectory depends on the opponent's reaction to the EV's own decision. It would be unrealistic to assume that the opponent's policy is known, and this fact represents a source of uncertainty. Following the approach from [13], we model the policy and dynamics of the opponent as a GP model, which is included in the EV controller.

In the following, we present the active exploration mechanism for head-to-head racing, which is an adaptation of the active exploration mechanism used in the time trial case. Here, the aim is to retrieve informative data from the opponent's reaction to several overtaking attempts of the EV. Further, we discuss specific limitations and challenges that pertain to the active exploration in head-to-head racing, due to the fact that the EV does not have full control over the feature space, and we outline how the active exploration takes place in the setting of a single competition with the opponent. In this racing scenario, we consider the *extended* state of the system $\boldsymbol{\xi}^{\mathrm{E}} = [\boldsymbol{\xi}, \boldsymbol{\xi}^{\mathrm{O}}]^\top$, which contains both the state of the ego vehicle $\boldsymbol{\xi}$ and of the opponent $\boldsymbol{\xi}^{\mathrm{O}}$. The opponent's state is defined as:

$$\boldsymbol{\xi}^{\mathrm{O}} = [s^{\mathrm{O}}, e_y^{\mathrm{O}}, e_\psi^{\mathrm{O}}, V_x^{\mathrm{O}}]^\top. \tag{13}$$

Here s^{O} and e_y^{O} are the longitudinal and lateral position of the opponent on the track, e_ψ^{O} is the yaw angle with respect to the reference of the track, and V_x^{O} is the longitudinal velocity.

GP Model: The combined effect of the opponent's dynamics and its policy is modeled as a GP, namely, we consider

$$y_k^{\mathrm{O}} = \boldsymbol{\xi}_{k+1}^{\mathrm{O}} = \boldsymbol{\xi}_k^{\mathrm{O}} + \boldsymbol{f}(\boldsymbol{\xi}_k^{\mathrm{O}}, \boldsymbol{\pi}_k(\boldsymbol{\xi}_k^{\mathrm{O}}, \boldsymbol{\xi}_k))T, \tag{14}$$

where $\boldsymbol{f}$ represents the real dynamics of the opponent and $\boldsymbol{\pi}$ represents the one-step opponent policy, which depends on the current opponent state $\boldsymbol{\xi}_k^{\mathrm{O}}$ and on the current EV state $\boldsymbol{\xi}_k$, in order to incorporate the opponent reaction to the EV decisions in the prediction steps [13]. We model the one-step closed-loop dynamics of the opponent in (14) with the GP

$$\boldsymbol{g}^{\mathrm{O}}(\boldsymbol{z}_{\mathrm{O}}) \sim \mathcal{N}\left(\boldsymbol{\mu}^{\mathrm{O}}(\boldsymbol{z}_{\mathrm{O}}), \boldsymbol{\Sigma}^{\mathrm{O}}(\boldsymbol{z}_{\mathrm{O}})\right), \tag{15}$$

where z_{O} is the GP input feature vector defined as:

$$\boldsymbol{z}_{\mathrm{O}} = [s^{\mathrm{O}} - s, e_y^{\mathrm{O}} - e_y, e_\psi, V_x, e_\psi^{\mathrm{O}}, V_x^{\mathrm{O}}, \bar{\boldsymbol{\kappa}}]^\top. \tag{16}$$

It contains the longitudinal and lateral distance between the EV and the opponent, the yaw angle and longitudinal velocity of both vehicles, and the vector $\hat{\boldsymbol{\kappa}}$ which contains the track curvature at a few look-ahead points, with a view to considering that humans typically choose their actions accounting for the future evolution of the track [13]. The GP input features (16) consist only of the relative configuration of the EV and of the opponents and of their position in the curvilinear Frenet coordinates rather than in the absolute coordinates, with a view to boosting the generalization capability of the GP prediction. Furthermore, the prediction of the opponent's trajectory is obtained by averaging over many samples from the GP model as in [13, Algorithm 1].

Active Exploration and MPC: The EV trajectory is computed iteratively by an MPC, based on the formulation in [13], where the cost function consists of several performance-based objectives. Unlike the time-trial case, the control objective in head-to-head racing is not to track an offline planned optimal trajectory over the track, which is infeasible to obtain in a prior, since the dynamic reaction of the opponent at run time must be considered. When adapting the active exploration mechanism we developed for the time trial, we aim to make minimal modifications to the baseline approach [13] for a close comparison. To this end, we introduce an additional term in the cost function that penalizes the deviation from a target reference state set for exploration, just as we did in the time trial. Specifically, the EV's target reference state $\boldsymbol{\xi}^{\mathrm{ref}}$ is selected to test the opponent's reaction to EV's overtaking attempts, in order to collect informative data for accurate opponent modeling. Since we no longer have access to an offline planned reference trajectory, we make the following adjustments to Algorithm 1 to determine the target reference states.

First, we determine the initial reference state $\boldsymbol{\xi}^{\mathrm{ref}}$ based on certain *nominal* behavior of the two vehicles: we approximate the one-step future relative configuration of the two vehicles, assuming they follow their current linear velocities and yaw angles. Then, such reference state $\boldsymbol{\xi}^{\mathrm{ref}}$ is modified using Algorithm 1. On the one hand, the distance of a candidate feature from the feature $\tilde{\boldsymbol{z}}$ visited following the nominal behavior is penalized, to prevent the EV from moving in a possibly dangerous way; on the other hand, visiting regions of the feature space with high posterior covariance is encouraged, to test the reaction of the opponent to behaviors of the EV for which the prediction of the reaction is more uncertain.

Given the target GP input feature vector $\boldsymbol{z}^{\mathrm{ref}}$ from Algorithm 1, the EV' reference state $\boldsymbol{\xi}_k^{\mathrm{ref}}$ is obtained with the following procedure: 1) The yaw angle e_ψ and the longitudinal velocity V_x are obtained from the third and fourth entry of $\boldsymbol{z}^{\mathrm{ref}}$, since $\boldsymbol{z}^{\mathrm{ref}}$ is defined as in (16). 2) We use the current GP model to predict the opponent's trajectory over the prediction horizon. Notably, the prediction is conditioned on a hypothetical future trajectory of the EV, which is the open-loop solution of the MPC problem at the last iteration. In doing so, we account for the opponent's reaction to the planned EV future movements without coupling the GP model with the optimization problem or the active exploration algorithm, as in [13]. 3) Then, the (time-varying) target longitudinal and lateral positions of the EV are obtained from the predicted trajectory of the opponent, subtracting the first and second entry of vector $\boldsymbol{z}^{\mathrm{ref}}$, respectively.

Note that the same target configuration associated with $\boldsymbol{z}^{\mathrm{ref}}$ is used to define the reference states for all the time steps over the prediction horizon. Ideally, we may select a different $\boldsymbol{z}^{\mathrm{ref}}$ for each time step to further maximize the exploration efficiency. However, estimating the posterior covariance of a feature vector at future time steps beyond the next one requires iteratively predicting the stochastic opponent reaction, which is computationally expensive. Meanwhile, the benefit can be marginal. In practice, the EV is less likely to reach the desired target configuration in one step, considering the trade-off in exploration and performance inherited in the MPC controller

and the uncertainty in the opponent's behavior. In light of these considerations, we set the EV's reference state $\boldsymbol{\xi}_k^{\text{ref}}$ for every prediction step based on the same $\boldsymbol{z}^{\text{ref}}$ to regularize the EV to stay close to the target configuration over the entire prediction horizon, aiming to make exploration more efficient while saving computational efforts.

Eventually, the MPC optimal control problem solved at each iteration is:

$$\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \alpha \sum_{k=1}^{N} \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^{\text{ref}}\|_{\boldsymbol{Q}}^2 + (1-\alpha)\Bigg(\sum_{k=0}^{N-1} q_c e_{y,k}^2 + \boldsymbol{u}_k^\top \boldsymbol{R} \boldsymbol{u}_k + \Delta \boldsymbol{u}_k^\top \boldsymbol{R}_d \, \Delta \boldsymbol{u}_k - q_s s_N^2 \Bigg) \tag{17a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k) T \quad \forall k = 0, \dots, N-1 \tag{17b}$$

$$s_0 = s(\boldsymbol{\xi}_t) \tag{17c}$$

$$s_{k+1} = s_k + V_{x,k} T \quad \forall k = 0, \dots, N-1 \tag{17d}$$

$$w_{\text{r},k} \leq e_{y,k} \leq w_{\text{l},k} \quad \forall k = 1, \dots, N \tag{17e}$$

$$\boldsymbol{u}_{\min,k} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max,k} \quad \forall k = 0, \dots, N-1 \tag{17f}$$

$$\boldsymbol{0} \geq \boldsymbol{h}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_k^{\text{O}}) \quad \forall k = 1, \dots, N. \tag{17g}$$

Other than the penalty for deviations with respect to the reference state $\boldsymbol{\xi}^{\text{ref}}$, weighted by matrix $\boldsymbol{Q} \geq 0$, cost function (17a) includes racing objectives from [13]: namely penalties for lateral offset from the center line e_y, and for large inputs and large rates of change of the input, where $q_c > 0$ and $\boldsymbol{R}, \boldsymbol{R}_d \geq 0$. Moreover, the last term is included to maximize the progress of the EV along the track depending on $q_s > 0$, where s is the longitudinal position along the track, initialized based on the current state $\boldsymbol{\xi}_t$ of the EV (17c) and predicted based on the predicted longitudinal velocity $V_{x,k}$ of the EV (17d), as in the baseline [13]. Alternatively, s could be handled as an independent optimization variable linked to the state [43]. The cost function is a convex combination of the reference tracking term and of the original racing term, ruled by parameter α. If $\alpha = 0$, the active exploration mechanism is completely disregarded and the cost function coincides with that from the baseline work [13], allowing a close comparison.

The EV dynamics (17b) is modeled as the dynamic bicycle model without compensation, as in this scenario, we focus exclusively on the uncertainty introduced by the unknown policy of the opponent to allow a close comparison with [13]. Constraints (17e) and (17f) enforce track boundary and input constraints, respectively. Constraint (17g) enforces collision avoidance with the opponent, whose predicted state $\boldsymbol{\xi}_k^{\text{O}}$ at step k is the GP prediction. The EV state reference $\boldsymbol{\xi}_k^{\text{ref}}$ also depends on the predicted opponent state $\boldsymbol{\xi}_k^{\text{O}}$ and on the selected GP input feature reference $\boldsymbol{z}^{\text{ref}}$ through the procedure outlined earlier in this subsection. Collision avoidance constraints also take the uncertainty of the prediction into account, as explained in the next subsection.

Probabilistic Collision Avoidance Constraints: Collision avoidance constraints (17g) consist of ellipsoidal regions around the predicted positions of the opponent that the EV must not enter. At first, the minimum covering ellipse given the physical dimensions of the opponent is considered; then, the ellipse is expanded by considering the uncertainty around the prediction of the opponent in longitudinal and lateral directions. Observe that the posterior covariance provided by the GP is fundamental to expanding the forbidden ellipsoidal regions. Finally, the constraints are implemented as soft constraints [13, Section IV], to allow for small violations of the expanded ellipsoidal regions if this yields a significant advantage in terms of performance, although such violation is disincentivized. More details on the collision-avoidance constraints are reported in [13, Section IV]. It is worth observing that, although the quadratic collision avoidance constraints make the optimal control problem non-convex, the solution is obtained efficiently using the `FORCESPRO` software [44].

Challenges and Limitations in Head-to-Head Racing: Relying on the target GP input feature selected via Algorithm 1 might not result in sufficiently diverse data. In fact, the opponent's future moves are not controlled by the EV, therefore the EV cannot arbitrarily enforce the future configuration of the two racing vehicles. It is possible that, while the EV is attempting to reach a given traffic configuration, the opponent reacts in a way to counterbalance the movement of the EV, and the configuration of the two reaches an equilibrium. In this case, although the absolute position of the two vehicles changes, the relative position does not change. If such equilibrium is reached and the change in the relative configuration is smaller than a threshold for several consecutive steps, we heuristically modify Algorithm 1 to encourage the exploration of GP input features corresponding to configurations of the two agents that are different from the current configuration by reversing vector $\boldsymbol{D}_i$ in Algorithm 1, with a view to breaking the stalemate. Furthermore, the target GP input feature is not updated for a few iterations to avoid reaching the same equilibrium.

Also, it is worth noting that, other than the GP input features that depend on both the EV and the opponent, namely the longitudinal distance, $s^{\text{O}} - s$, and the lateral distance, $e_y^{\text{O}} - e_y$, there exist several GP input features over which the EV has no control, i.e., the opponent's yaw angle and lateral velocity and the curvature of the look-ahead points. Exploration cannot be encouraged for such features. Therefore, when generating the list of candidate GP input features $\{\boldsymbol{z}_{\text{O}}^{(i)}\}_i^{n_{\text{G}}}$ for Algorithm 1, we consider features that differ only in the components over which the EV has influence. Otherwise, the algorithm might choose a target GP input feature $\boldsymbol{z}_{\text{O}}^{\text{ref}}$ because of the high posterior covariance given by the curvature value, for example, which the EV cannot impose, and possibly resulting in a relative configuration for which the posterior covariance is already small. Focusing only on the input features over which the EV has influence is beneficial also to maintain the number of candidate target GP input features n_{G} limited.

Iterative Framework in Head-to-Head Racing: Finally, we outline how our iterative scheme works for the scenario with the opponent. At first, an initial GP model is used during the exploration phase. At this stage, it is not required that the GP model is accurate since it will improve in the training after the exploration phase. Nevertheless, a GP model is needed, as the regions of the feature space that must be explored are chosen using the posterior covariance. Therefore, a coarse model is trained at first, possibly from data collected with other opponents, and therefore not tailored for the current opponent.

Then, the exploration phase takes place, which starts with

a high value of α in the EV optimal control problem (17). The goal of this phase is a trade-off between winning the race and collecting a variety of informative data points about the opponent's policy in reaction to several attempts of the EV. At the end of the exploration phase, which can last for a few minutes, the GP is retrained on a remote platform, while the EV continues the competition. As soon as the training of the updated GP model has been completed, the updated GP model is transferred to the EV, which now can leverage an accurate prediction of the opponent's behavior and focus on winning the race, that is, setting parameter $\alpha = 0$ in problem (17).

IV. Simulations

In this section, we describe the simulations that were conducted to validate our iterative GP regression framework with active exploration mechanisms in both autonomous racing scenarios. In the time trial, Section IV-A, we compare performance with previous work [14], where a double iterative GP regression framework is used without active exploration. We show that our approach yields an improvement in the minimum lap time as a result of the more accurate learning performance that the enriched dataset from the exploration allows. Then, in the challenge with the opponent, Section IV-B, we compare our approach with the approach from [13], in which a GP predictor of the future trajectory of the opponent is trained on a large dataset of shorter runs. Notably, our approach results in an improvement in the average EV performance as a consequence of the improved prediction of the opponent for further prediction steps, although our approach relies on a significantly smaller dataset of measurements collected during a single phase of exploration.

A. Time Trial

We used the same simulation setup as in previous work [14] for an accurate and fair comparison. The closed-loop simulations were carried out in the highly realistic racing simulation platform Gran Turismo Sport from Sony Interactive Entertainment Inc [15], using as EV the Audi TT Cup running on the Tokyo Expressway Central Outer Loop Track. The desktop computer wired connected to the Play Station 5 is an Alienware-R13, with CPU Intel i9-12900 and GPU Nvidia 3090. Our code is developed in Python. The QP MPC optimal control problem (9) is solved using `qpsolvers` [45]. The MPC frequency is 20 Hz and the prediction horizon $N = 20$. The nonlinear optimal control problem for path planning (18) is solved using `CasADi` [46], [47]. The vehicle parameters are reported in Table I. The GP regression is implemented using `GPyTorch` [48], which exploits the GPU and adopts an efficient and general approximation of GPs based on black-box matrix-matrix multiplication. Over 10,000 data points are supported in the GP dataset while preserving almost the same prediction accuracy and making the GP regression estimation feasible in real-time.

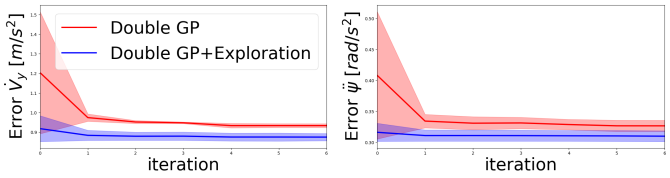
In our implementation, the maximum size of the dataset is $M = 2000$. The risk parameter in the tightened constraints (10) is $\beta = 0.6$. It is important to observe that since the target GP input feature is changed at each iteration of the MPC algorithm, we collect data points also about sudden changes in the features, which are relevant for the GP model in the planning problem (18). To speed up the target GP input feature selection online in Algorithm 1, we evaluate and store the covariance of each point in the list of candidate features $\{\boldsymbol{z}^{(i)}\}_i^{n_\mathrm{G}}$ before the start of each trial, after retraining the GP. Inspired by the discussion on the value of α provided in the previous work [39], during the first two iterations, we encourage the exploration setting $\alpha_0 = \frac{6}{7} = 0.857$ and $\alpha_1 = \frac{5}{7} = 0.714$. These values were chosen to linearly decrease α to zero in 7 steps, to completely stop the exploration mechanism in 7 rounds. However, the results suggested that after just a few rounds of exploration, the dataset of selected diverse measurements would not change significantly because the collected measurements are sufficiently diverse, making further exploration not helpful. Therefore, from the third iteration, α is set to zero, instead of gradually decreasing it to zero, that is, from iteration number 2, the focus is exclusively on minimizing the lap time.

TABLE I: Parameters of Audi TT Cup in GTS

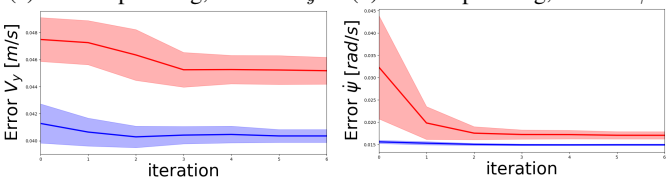
Parameter	Value
Total mass m	1161.25 kg
Length from CoG to front wheel l_f	1.0234 m
Length from CoG to rear wheel l_r	1.4826 m
Width of chassis	1.983 m
Height of CoG h_c	0.5136 m
Friction ratio μ	1.5
Wind drag coefficient C_{xw}	0.1412 kg/m
Moment of inertia I_{zz}	2106.9543 Nm

In the first run, the EV uses the nominal MPC, without the GP compensation, to track a curvature-optimal path [17], and the measurements are used to train the two GPs. In the following iterations, the GPs are exploited and the time-optimal path is re-planned and tracked. The planning problem is warm-started with the planned trajectory from the previous iteration to discourage large deviations from the previous trajectory since this could result in planning infeasible trajectories. Furthermore, we have heuristically observed that the optimal planned trajectory does not improve significantly after the first iteration of the optimization, therefore we only run one iteration. The iterative framework has been repeated for 7 iterations. Because of small uncertainties in the timing of the communication network between the computer implementing our algorithm and the simulation environment in the Play Station, slight variations are observed when the same simulation is repeated. Thus, we have repeated each simulation three times, considering the mean of the measured times and the standard deviation between the three trials.

First, we evaluate the learning performance over the iterations, both for the GP used in the planning and for the GP used in the tracking phase. The evaluation, shown in Figure 2, investigates the prediction error over a dataset of diverse measurements, collected during a run in which the EV repeatedly deviated from the center line of the track. With respect to the previous work [14], the diverse dataset collected during the active exploration allows a reduction of

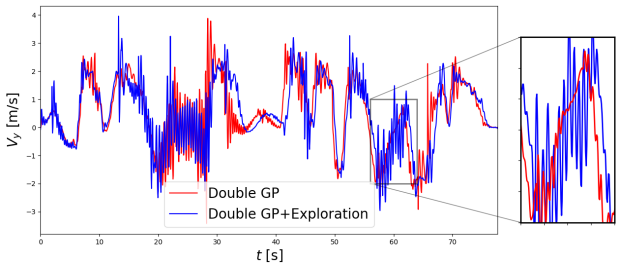


(a) GP for planning, feature $\dot{V}_y$. (b) GP for planning, feature $\ddot{\psi}$.

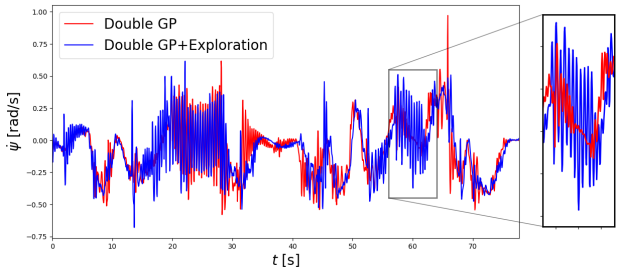


(c) GP for tracking, feature V_y. (d) GP for tracking, feature $\dot{\psi}$.

Fig. 2: Prediction error between compensated model and data of the dynamics, for each iteration in *Time-Trial* task. Each simulation has been repeated for three trials: solid lines indicate the mean, and the shaded areas represent the standard deviation across the three trials.



(a) Data collected for V_y.



(b) Data collected for $\dot{\psi}$.

Fig. 3: State analysis of *Time Trial* simulations in Gran Turismo Sport. Data was collected in the first iteration, in which the EV dynamics is tested by repeatedly deviating from the trajectory of the planned path ($\alpha_0 = 6/7$).

the prediction error for both the GP used in the planning and the GP used in the tracking phase.

Figure 3 presents the data collected in the first iteration. Both from the signal of the longitudinal velocity V_y, Figure 3a, and the signal of the derivative of the yaw angle $\dot{\psi}$, Figure 3b, we observe that during the run the EV dynamics is tested by repeatedly deviating from the trajectory of the planned path.

In conclusion, we discuss the advantages of our method evaluating the lap time of the optimal planned trajectory and the recorded lap time during the experiments, since the GP compensates the nominal dynamics both at run time and in the planning phase. Figure 4 shows the lap times obtained with the comparison with the **Double GP** method, the previous approach [14], and our proposed **Double GP+Exploration** which includes our active exploration method in Algorithm 1. For convenience, iteration 0 is the first trial in which the GP compensation is used, that is, neglecting the curvature-optimal run. As shown in Figure 4a, in our approach, the time of the optimal planned path increases over the iterations, although the dispersion within repeated simulations decreases. This is understood as a consequence of the improvement in learning performance. In fact, the goal of the path planner is to derive the optimal path that is feasible for the actual dynamics of the EV, therefore it is reasonable that a more accurate dynamics results in an optimal planned path with higher lap time. Considering the actual lap time measured at each iteration, reported in Figure 4b, we first observe a significant increase in the lap time yielded by our algorithm, which is due to the fact that in the first two iterations, the exploration takes place, and therefore the performance objectives are partially compromised to collect diverse measurements. However, from iteration 2, the focus of the EV is on minimizing the lap time and we observe a decrease in the minimum time as well as in the deviation between repeated simulations compared to the baseline [14]. Finally, the difference between the lap time of the planned path and the actual lap time of the run, Figure 4c, shows that the diverse dataset resulting from the active exploration reduces the gap between the time of the planned path and the actually achieved minimum time over the iterations.

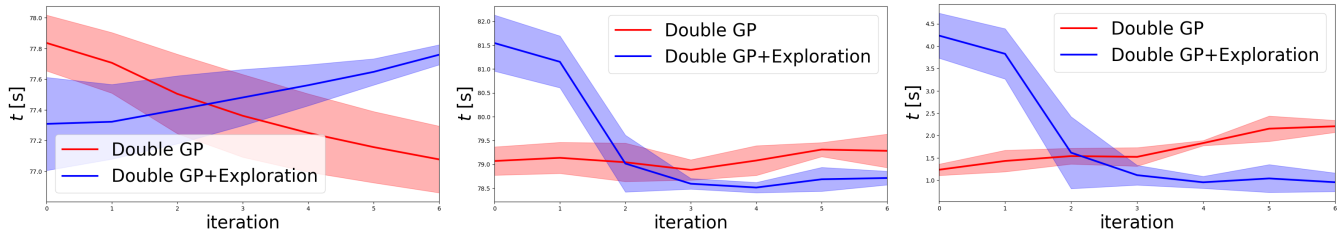
B. Head-to-head Racing

For the competition against the opponent, we use the simulation setup from [13], which implements a racing environment for miniature racing cars[1] The control frequency is 10 Hz, and the prediction horizon is $N = 10$. All simulations are run on a laptop with an AMD Ryzen 5 3500U eight-core processor.

Vector $\boldsymbol{\kappa}$ in the GP input feature (16) consists of the curvature at three look-ahead points, to facilitate the comparison with [13]. In all simulations, the opponent is implemented as an MPC-controlled agent with a blocking policy [12]. Other than performance-based objectives, the cost function of the opponent penalizes deviations from the current lateral position on the track of the EV, so that the opponent "mirrors" the EV lateral behavior and blocks overtaking attempts. To encourage overtaking attempts of the EV, the parameter q_s ruling the progress maximization reward of the EV in (17a) is set higher than for the TV. Further details are given in [13].

We compare two methods for modeling the opponent with the GP. In **Baseline** [13], the GP model for the opponent is trained using closed-loop trajectories from an offline dataset of 500 runs in which the EV starts behind the opponent on randomly generated tracks. In contrast, our proposed **Data Selection + Exploration** method uses the iterative and exploration-based approach presented in Section III-B. For the initial GP model, we use a *smaller* initial offline dataset of

[1]https://github.com/MPC-Berkeley/gp-opponent-prediction-models. The optimal control problem (17) is solved using the FORCESPRO software [44].



(a) Time of the planned path. (b) Time measured during closed-loop test. (c) Gap between planned and measured times.

Fig. 4: Lap times obtained in each iteration in *Time Trial* simulations in Gran Turismo Sport. Each simulation has been repeated for three trials: solid lines indicate the mean over the three trials, and the shaded areas represent the standard deviation across the three trials. At each iteration, **Double GP+Exploration** uses $\alpha_0 = 6/7$, $\alpha_1 = 5/7$, and $\alpha_{2+} = 0$ to transition from high exploration to no exploration.

20 runs, generated with the same mechanism as the Baseline method. This allows us to test how well the GP model can be improved during the exploration. The exploration phase lasts 10 minutes of simulation time and is run on the closed track provided by [13]. During exploration, we set $\alpha = 0.9$, so that the EV focuses primarily on testing the opponent's reaction to several EV movements. We simulate only one exploration and retrain phase, thus the GP model is retrained only at the end of the exploration. Then, the parameter α is set to zero, that is, the EV focuses exclusively on winning the race.

To test each method, we randomly generate a set of 100 scenarios. A track is randomly generated from straight, curved, and chicane stretches, and the length and curvature of each stretch are randomly selected. For each track, the initial longitudinal position and velocity of the cars are randomly generated, but the EV is always behind, to test the overtaking ability. Each simulation is interrupted 1.5 seconds after overtaking occurs, or when the EV reaches the track end.

We summarize the results in Table II. In 100 simulations, no major collision is observed—the EV never leaves the track or crashes during the trials. Nevertheless, the EV hits the track border in 2 simulations when the GP predictor from [13] is used, and 5 times when using our GP trained with the exploration data. These minor collisions could be prevented by adding safety margins to the constraints, at the price of compromising the performance. For the average overtaking time, we consider the simulations on the 93 tracks in which the EV stays strictly inside the track boundaries with both predictors. On average, the EV overtakes the opponent 0.33 seconds earlier when using the GP trained on the exploration data compared to the baseline approach. It should be observed that this improvement in the EV performance is achieved with a significantly smaller training dataset, that is, roughly 600 data points collected during the exploration, as opposed to the dataset of sample runs used to train the baseline GP [13], consisting of roughly 5000 data points. The video available at https://youtu.be/gaAuzsR2fII shows the exploration phase in the challenge with the opponent and the transition to focusing on winning the race after the training of the GP with the enriched dataset.

Finally, we analyze the GP prediction accuracy to test how well the data selection and exploration mechanisms reduce model error. We compare the methods to a third GP, the **Data Selection** method, which trains a GP on a small dataset of the most diverse measurements within the dataset used for the Baseline GP, therefore without exploration. The dataset is selected using the procedure outlined in Section III. We evaluate the impact of employing only the most diverse data points collected within several runs, without employing the active exploration mechanism in the data collection. We assess model accuracy by comparing the prediction error of the lateral position of the opponent, which is of primary concern for overtaking maneuvers. We perform the analysis offline using the 200 closed-loop trajectories of the opponent collected from the simulations on the 100 tracks in which the EV first uses the Baseline GP and then our GP with Data Selection and Exploration. Since the trajectory of the opponent depends on the EV's own behavior, we repeat the prediction offline using data from all 200 trajectories for all three GP predictors, for a fair comparison.

The analysis of the accuracy is shown in Figure 5, using the data from one of the 200 closed-loop opponent trajectories. Furthermore, on the right of Table II we report the average results over all 200 closed-loop trajectories. Each GP method has a comparable accuracy in the 1-step-ahead prediction of the lateral position of the opponent, shown in Figure 5a. However, there are occasional spikes in prediction error, especially with the GP only using Data Selection to improve dataset diversity. This likely indicates that exploration is necessary to improve the diversity of the dataset and, thus, the accuracy of the GP.

While 1-step prediction accuracy is important, accurately predicting the opponent's behavior over *long* time horizons is also very important, given the fact that the EV's behavior is predicted using an N-step prediction horizon in the MPC. Thus, we compare each method's 9-step prediction accuracy in Figure 5b. Our proposed GP with data selection and exploration significantly outperforms the other methods, resulting in a smaller 9-step prediction error compared to the two other predictors. In fact, comparing the average t-step prediction error in Figure 5c as a function of the number of steps in the horizon t, we see that the exploration mechanism decreases modeling error compared to the Baseline GP or Data Selection alone. As expected, each method's accuracy deteriorates for further prediction steps; however, our exploration-based GP results in the slowest increase in the mean and standard deviation of prediction error as the prediction horizon

TABLE II: *Head-to-Head* racing results over 100 tracks

GP Predictor type	Hit track border	Average overtaking time mean±std [s]	Prediction error mean±std [m] over all simulations 1-step-ahead	2-step-ahead	8-step-ahead	9-step-ahead
Baseline [13]	2	12.772±4.353	0.003±0.003	0.009±0.009	0.100±0.100	0.119±0.119
Data Selection + Exploration	5	12.442±5.041	0.004±0.004	0.007±0.007	0.046±0.046	0.055±0.055
Data Selection[1]	/	/	0.005±0.005	0.011±0.011	0.108±0.105	0.131±0.127

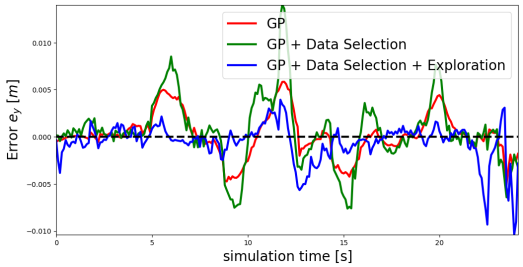
[1] The *Data Selection* GP predictor is only used for the offline analysis of the prediction accuracy. Therefore, we only report the average prediction errors computed in the offline analysis with respect to the closed-loop opponent trajectories.

increases. Since the Baseline GP and GP with Data Selection achieve similar modeling performance, this indicates that the exploration mechanism can make a notable improvement in the training dataset by purposefully opting to collect data in regions with greater modeling uncertainty. Thus, with the greater long-step prediction accuracy of the opponent's model, our GP with Data Selection and Exploration improves the performance of the EV, since the strategic decisions especially rely on the prediction for further steps.

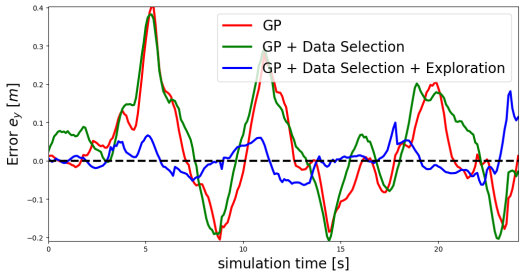
V. Discussion

An important aspect regarding active exploration, especially in head-to-head racing, is the criterion to terminate the exploration phase. The decision should be grounded on the improvement in GP's prediction accuracy. However, assessing the prediction accuracy requires retraining the GP with the updated dataset, which is time-consuming and, although it could be possible in principle between different runs for the time trial, it is not feasible in real time for head-to-head racing. We instead assess the diversity of collected measurements. During operation, the dataset is incrementally extended with new measurements following the procedure outlined in Section III-A. In practice, the dataset reaches a steady state after a few minutes of exploration, after which most new measurements are discarded as they no longer enhance data diversity. Thus, the data update frequency serves as a good heuristic to determine when to conclude the exploration phase. Meanwhile, the update rate is also used to guide the systematic scheduling of the exploration weight, α, during exploration: higher rates suggest that the dataset does not well represent the feature space, thus exploration should be encouraged; instead, lower rates suggest the dataset is sufficiently diverse, thus exploration should be discouraged, and α should gradually decay to zero to focus on winning the race.

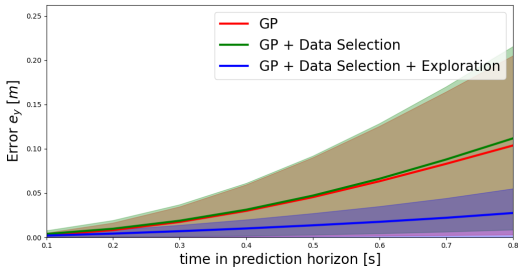
Potentially, multiple rounds of exploration with more fine-grained data update strategy and α scheduling can be investigated to further optimize the exploration procedure. However, it should be noted that the exploration and following training phase must not take too long with respect to the duration of the competition. Otherwise, too little time remains to focus on taking advantage of the improved model to win the race. Our simulation results show that a single phase of exploration with the simple termination strategy described above already yields a significant improvement, as we discuss in Section IV-B.



(a) 1-step-ahead prediction during the run.



(b) 9-step-ahead prediction error during the run.



(c) Average prediction error t-seconds-ahead. Solid lines indicate the mean, the shaded areas the standard deviation.

Fig. 5: Prediction error of the opponent lateral position e_y with respect to the same closed-loop trajectory from one of the closed-loop trajectories opponent.

Furthermore, the framework proposed in this work implicitly assumes that the behavior of the opponent and its reaction to the EV is stationary—which is a common assumption in literature [13]—and thus can be accurately learned with data collected in finite exploration time. If the opponent is similarly featured with exploration capabilities of the EV behavior, this might not be the case, since the opponent policy will keep changing. One possibility to deal with a time-varying

opponent policy is to run several phases of exploration, then exploitation, and re-exploration when the prediction error grows again. However, in the presence of an opponent with symmetric exploration capabilities, this strategy might result in a deadlock. A thorough discussion and a practical solution to this problem are beyond the scope of this work, which we leave as potential future extensions.

Finally, although we introduce a GP compensation term in the hope of capturing the dynamic effects not modeled by the nominal bicycle model, the feature space of the GP is still defined from the state space of the bicycle model. As a result, certain unmodeled phenomena are not able to be captured by the GP model due to some missing states, for example, the roll and pitch angles of the vehicle. Nevertheless, we still managed to achieve significant improvement compared to pure physics-based models, showing the promising potential of our GP-based compensation model and active exploration scheme. To further enhance the model's accuracy, one potential solution to account for the missing states is to add the history of the observations to the GP's inputs so that the model can learn to implicitly infer the unobservable states, which we plan to explore in our future work.

VI. Conclusion

In this work, we presented an iterative Gaussian Process regression scheme for autonomous racing implementing an active exploration mechanism. During the first iterations, the EV trajectory is planned trying to collect measurements for the states with high posterior covariance. Among the collected measurements, a smaller dataset is obtained, selecting the most diverse data points, and is used to retrain the Gaussian Process model. Then, in further iterations of the algorithm, the focus is exclusively on improving the performance of the EV, leveraging on the improved prediction accuracy. We showed that the GP exploration method can be applied both when the GP model is used for error compensation and for opponent modeling. We tested the framework to compensate for the modeling errors in the EV dynamics near handling limits in Gran Turismo Sport [15], and to model the opponent's reaction to the EV's own decisions in a simulation environment. In both scenarios, we obtained a significant improvement in the prediction accuracy and, consequently, in the EV performance.

Future research will focus on validating the active exploration approach for the opponent challenge in other simulation environments, with a special focus on the generalizability of the approach to different opponent policies. In addition, theoretical guarantees that the iterative framework improves the accuracy of the GP should be researched. Furthermore, the framework will be tested in scenarios where both uncertainties, namely modeling errors in the vehicle dynamics and the opponent policy, are tackled simultaneously. Moreover, strategies to address scenarios in which the opponent policy is non-stationary and possibly also implementing an exploration strategy should be investigated.

Another interesting direction for future research stems from the fact that exploration mechanisms relying on the measure of uncertainty provided by the GP tend to generate uniform experiment designs. However, the ultimate goal of the control is to drive the EV optimally, not purely minimize the model uncertainty. In light of this, new exploration strategies not based on heuristics should be developed and tested.

Acknowledgments

The authors thank Shaoshu Su for valuable discussions, and Ce Hao for the simulation setup in GTS. We also would like to thank Kenta Kawamoto from Sony AI for his kind help and fruitful discussions. This work was supported by Sony AI, and Polyphony Digital Inc., which provided the Gran Turismo Sport framework. T. Benciolini's visit to the University of California, Berkeley, was supported by a fellowship within the "Research Grants for Doctoral Students" program of the German Academic Exchange Service (DAAD) and the Bavaria California Technology Center (BaCaTeC) grant 12-[2022-2]. Catherine Weaver is supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1752814. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Appendix

Planning Optimal Control Problem

We transform the integral in time into an integral in the spatial domain [49] and closely approximate the solution that would be obtained by directly considering the lap time as objective, as in [17], where a more complete discussion is provided. The time-optimal trajectory is obtained by solving

$$\min_{\{\boldsymbol{u}_k\}_{k=0}^{N_p-1}} \sum_{k=0}^{N_p-1} \frac{(1-\kappa(s_k)e_{y,k})\Delta s_k}{V_{x,k}\cos(e_{\psi,k}) - V_{y,k}\sin(e_{\psi),k}} \tag{18a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k)T + \boldsymbol{\mu}^{\text{plan}}(\boldsymbol{z}_{\text{plan}}(\hat{\boldsymbol{\xi}}_k, \hat{\boldsymbol{u}}_k)), \quad \forall k = 0, \dots, N_p - 1 \tag{18b}$$

$$\boldsymbol{\xi}_{N_p} = \boldsymbol{\xi}_0 \tag{18c}$$

$$w_{\text{r},k} \leq e_{y,k} \leq w_{\text{l},k}, \qquad \forall k = 1, \dots, N_p \tag{18d}$$

$$\delta_{\min,k} \leq \delta_k \leq \delta_{\max,k}, \qquad \forall k = 0, \dots, N_p - 1, \tag{18e}$$

where N_p is the length of the horizon for the planning problem. Cost function (18a) results from the transformation of the time-optimal objective into an integral in the spatial domain, as in [17], and Δs_k is the incremental longitudinal progress along the path. Constraint (18c) enforces that the path starts and ends in the same point, whereas (18d) and (18e) guarantee that the trajectory does not leave the right and left track boundaries w_{r} and w_{l} and that the steering angle δ remains in the actuation range $[\delta_{\min}, \delta_{\max}]$. Constraint (18b) guarantees that the planned trajectory is feasible for the vehicle dynamics, which is crucially important to ensure that the vehicle can track the optimal trajectory. $\boldsymbol{\mu}^{\text{plan}}(\boldsymbol{z}_{\text{plan}}(\hat{\boldsymbol{\xi}}_k, \hat{\boldsymbol{u}}_k))$ represents the modeling error compensation provided by the GP model, correcting the inaccuracies of the physics-based

model when the vehicle is driven at handling limits. We model the dynamics compensation as a GP

$$\boldsymbol{g}^{\text{plan}}(\boldsymbol{z}_{\text{plan}}) \sim \mathcal{N}\left(\boldsymbol{\mu}^{\text{plan}}(\boldsymbol{z}_{\text{plan}}), \boldsymbol{\Sigma}^{\text{plan}}(\boldsymbol{z}_{\text{plan}})\right). \tag{19}$$

We compensate the two states with the greatest impact on the prediction error, namely $\dot{V}_y$ and $\ddot{\psi}$ [14], therefore $\boldsymbol{\mu}^{\text{plan}}(\boldsymbol{z}_{\text{plan}}) = [0, \mu^{\text{plan}}_{\dot{V}_y}(\boldsymbol{z}_{\text{plan}}), \mu^{\text{plan}}_{\ddot{\psi}}(\boldsymbol{z}_{\text{plan}}), 0, 0, 0]^\top$ [3]. As in [14], we consider as input features of the planning GP $\boldsymbol{g}^{\text{plan}}$ vector $\boldsymbol{z}_{\text{plan}} = [V_y, \dot{\psi}, \delta]^\top$, being the most correlated with the output features. In order not to embed the GP model into the optimization problem, the input feature $\boldsymbol{z}_{\text{plan}}$ is defined with respect to nominal state and input vector $\tilde{\boldsymbol{\xi}}_k$ and $\tilde{\boldsymbol{u}}_k$, that is, the solution of the previous iteration of the optimization [17], rather than on the actual state and input.

REFERENCES

[1] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[2] R. Rajamani, *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[3] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with Gaussian Process Dynamics for Autonomous Miniature Race Cars," in *2018 European Control Conference (ECC)*, 2018.

[4] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning Model Predictive Control," in *2017 American Control Conference (ACC)*, 2017.

[5] U. Rosolia and F. Borrelli, "Learning How to Autonomously Race a Car: A Predictive Control Approach," *IEEE Transactions on Control Systems Technology*, 2020.

[6] L. P. Fröhlich, C. Küttel, E. Arcari, L. Hewing, M. N. Zeilinger, and A. Carron, "Model Learning and Contextual Controller Tuning for Autonomous Racing," 2021.

[7] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under Uncertainty in Sparse Spectrum Gaussian Processes with Applications to Filtering and Control," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 2760–2768.

[8] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürr, "Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, 2021.

[9] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "PiP: Planning-Informed Trajectory Prediction for Autonomous Driving," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Springer International Publishing, 2020.

[10] J. Liu, W. Zeng, R. Urtasun, and E. Yumer, "Deep Structured Reactive Planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[11] Z. Wang, R. Spica, and M. Schwager, "Game Theoretic Motion Planning for Multi-robot Racing," in *Distributed Autonomous Robotic Systems*, ser. Springer Proceedings in Advanced Robotics, N. Correll, M. Schwager, and M. Otte, Eds. Springer International Publishing, 2019.

[12] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian Process-based Stochastic Model Predictive Control for Overtaking in Autonomous Racing," in *International Conference on Robotics and Automation (ICRA), Workshop on Opportunities and Challenges with Autonomous Racing*, 2021.

[13] E. L. Zhu, F. Lukas Busch, J. Johnson, and F. Borrelli, "A Gaussian Process Model for Opponent Prediction in Autonomous Racing," 2022.

[14] S. Su, C. Hao, C. Weaver, C. Tang, W. Zhan, and M. Tomizuka, "Double-Iterative Gaussian Process Regression for Modeling Error Compensation in Autonomous Racing," 2023.

[15] S. I. E. Inc., "Gran Turismo Sport," [Online]. Available: https://www.gran-turismo.com/us/gtsport/top/.

[16] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[17] C. Hao, C. Tang, E. Bergkvist, C. Weaver, L. Sun, W. Zhan, and M. Tomizuka, "Outracing Human Racers with Model-based Autonomous Racing," 2022.

[18] H. Pacejka, *Tire and Vehicle Dynamics*. Elsevier, 2005.

[19] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-Based Model Predictive Control for Autonomous Racing," *IEEE Robotics and Automation Letters*, 2019.

[20] A. Jain, M. O'Kelly, P. Chaudhari, and M. Morari, "BayesRace: Learning to race autonomously using prior experience," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, 2021.

[21] A. Wischnewski, J. Betz, and B. Lohmann, "A Model-Free Algorithm to Safely Approach the Handling Limit of an Autonomous Racecar," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019.

[22] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, "A Game-Theoretic Approach to Replanning-Aware Interactive Scene Prediction and Planning," *IEEE Transactions on Vehicular Technology*, 2016.

[23] A. Dreves and M. Gerdts, "A generalized Nash equilibrium approach for optimal control problems of autonomous cars," *Optimal Control Applications and Methods*, 2018.

[24] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, "Multi-Hypothesis Interactions in Game-Theoretic Motion Planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[25] B. Evens, M. Schuurmans, and P. Patrinos, "Learning MPC for Interaction-Aware Autonomous Driving: A Game-Theoretic Approach," in *2022 European Control Conference (ECC)*, 2022.

[26] J. L. V. Espinoza, A. Liniger, W. Schwarting, D. Rus, and L. V. Gool, "Deep Interactive Motion Prediction and Planning: Playing Games with Motion Prediction Models," in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*. PMLR, 2022.

[27] M. Li and I. Sethi, "Confidence-based active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[28] R. Rickenbach, J. Köhler, A. Scampicchio, M. N. Zeilinger, and A. Carron, "Active Learning-based Model Predictive Coverage Control," *IEEE Transactions on Automatic Control*, pp. 1–16, 2024.

[29] D. Silver, J. A. Bagnell, and A. Stentz, "Active learning from demonstration for robust autonomous navigation," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 200–207.

[30] I. Abraham and T. D. Murphey, "Active Learning of Dynamics for Data-Driven Control Using Koopman Operators," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1071–1083, 2019.

[31] S. Müller, A. von Rohr, and S. Trimpe, "Local policy search with Bayesian optimization," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 20 708–20 720.

[32] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Recent Advances in Bayesian Optimization," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 287:1–287:36, 2023.

[33] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.

[34] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, "Active learning in robotics: A review of control principles," *Mechatronics*, vol. 77, p. 102576, 2021.

[35] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A Survey of Deep Active Learning," *ACM Computing Surveys*, vol. 54, no. 9, pp. 180:1–180:40, 2021.

[36] J. Park and K. H. Law, "Bayesian Ascent: A Data-Driven Optimization Scheme for Real-Time Control With Application to Wind Farm Power Maximization," *IEEE Transactions on Control Systems Technology*, 2016.

[37] S. Bin-Karim, A. Bafandeh, A. Baheri, and C. Vermillion, "Spatiotemporal Optimization Through Gaussian Process-Based Model Predictive Control: A Case Study in Airborne Wind Energy," *IEEE Transactions on Control Systems Technology*, 2019.

[38] A. Siddiqui, J. Borek, and C. Vermillion, "A Fused Gaussian Process Modeling and Model Predictive Control Framework for Real-Time Path Adaptation of an Airborne Wind Energy System," *IEEE Transactions on Control Systems Technology*, 2023.

[39] S. Yang, N. Wei, S. Jeon, R. Bencatel, and A. Girard, "Real-time optimal path planning and wind estimation using Gaussian process regression for precision airdrop," in *2017 American Control Conference (ACC)*, 2017.

[40] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[41] D. Nguyen-Tuong and J. Peters, "Incremental online sparsification for model learning in real-time robot control," *Neurocomputing*, 2011.

[42] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *12th International Symposium on Advanced Vehicle Control*, 2014.
[43] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.
[44] A. Domahidi and J. Jerez, "FORCES Professional," Embotech AG, 2014.
[45] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *2013 European Control Conference (ECC)*, 2013.
[46] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2019.
[47] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, 2006.
[48] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration," in *Advances in Neural Information Processing Systems*, 2018.
[49] N. R. Kapania, J. Subosits, and J. Christian Gerdes, "A Sequential Two-Step Algorithm for Fast Generation of Vehicle Racing Trajectories," *Journal of Dynamic Systems, Measurement, and Control*, 2016.

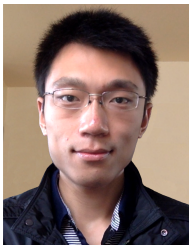
Tommaso Benciolini received the M.Sc. Degree in Automation Engineering from the University of Padova, Italy, in 2020. During his studies, he joined the Control and Power Research Group of the Department of Electrical and Electronic Engineering of the Imperial College of London, UK. In 2020 he joined the Chair of Automatic Control Engineering at the Technical University of Munich, Germany, as a research associate and Ph.D. student. In 2023 he was a visiting researcher at the Mechanical Systems Control Lab at the University of California, Berkeley, USA. His research interests include advances in Model Predictive Control with applications in automated driving.

Chen Tang (Member, IEEE) received his PhD in Mechanical Engineering at UC Berkeley in 2022. Prior to this, he received his bachelor's degree in mechanical engineering from the Hong Kong University of Science and Technology (HKUST). He is currently a postdoctoral fellow in the Department of Computer Science at UT Austin. His research interest lies at the intersection of control, robotics, and learning. The goal of his research is to develop trustworthy and safe autonomous agents interacting with humans. He won second place in the IEEE ITSC 2018 Best Student Paper Award and the AMSE DSCD Rising Star 2022 Award. He was selected as an RSS Pioneer in 2023.

Marion (nee Sobotka) Leibold received the Diploma degree in applied mathematics (2002) from Technical University of Munich, Germany. Further, she received the Ph.D. (2007) and Habilitation (2020) in control theory from Technical University of Munich, Germany. She is currently a Senior Researcher with the Faculty of Electrical Engineering and Information Technology, Institute of Automatic Control Engineering, Technical University of Munich. Her research interests include optimal control and nonlinear control theory, and the applications to robotics and automated driving.

Catherine Weaver received her M.S. in mechanical engineering from University of California, Berkeley, CA, USA in 2021. Prior to that she received a B.S. in mechanical engineering from Purdue University in West Lafayette, IN, USA in 2019. She is currently pursuing a Ph.D in the Department of Mechanical Engineering at the University of California, Berkeley, CA, USA. Her research interest is in advanced control and machine learning for autonomous driving.

Masayoshi Tomizuka (Life Fellow, IEEE) received the Ph.D. degree in mechanical engineering from MIT in February 1974. In 1974, he joined as the Faculty Member of the Department of Mechanical Engineering at the University of California at Berkeley, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair. His current research interests include optimal and adaptive control, digital control, signal processing, motion control, and control problems related to robotics, precision motion control and vehicles. He was the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of NSF from 2002 to 2004. He was a Technical Editor of the ASME Journal of Dynamic Systems, Measurement and Control (J-DSMC) from 1988 to 1993, and the Editor-in-Chief of the IEEE/ASME TRANSACTIONS ON MECHATRONICS from 1997 to 1999. He is a Fellow of the ASME and IFAC. He was a recipient of the Charles Russ Richards Memorial Award (ASME), in 1997, the Rufus Oldenburger Medal (ASME), in 2002, and the John R. Ragazzini Award in 2006.



Wei Zhan (Member, IEEE) received the Ph.D. degree from University of California, Berkeley, in 2019. He is currently an Assistant Professional Researcher with UC Berkeley, and the Co-Director of Berkeley Deep Drive Center. His research interests include targeting scalable and interactive autonomy at the intersection of computer vision, machine learning, robotics, and control and intelligent transportation. His publications received the Best Student Paper Award in IV 2018 and the Best Paper Award–Honorable Mention in IEEE ROBOTICS AND AUTOMATION LETTERS. He is the Lead Author of the INTERACTION dataset, and organized its prediction challenges in NeurIPS 2020 and ICCV 2021.