2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Computation-Aware Data-Driven Model Discrimination with Application to Driver Intent Identification

Mohit Bhagwat, Zeyuan Jin, Sze Zheng Yong

Abstract— **In this paper, we consider the problem of designing a model discrimination algorithm for partially known systems, where only sampled data of the unknown dynamics are available. Leveraging data-driven abstraction methods to over-approximate the unknown dynamics and an incremental abstraction approach, we propose a method to find a pair of piecewise affine functions that "includes" all possible trajectories of the original unknown dynamics, which further simplify the data-driven abstraction and would scale better for high dimensional systems. Then, using the models from the abstraction method, we analyze the detectability of these models from noisy, finite data as well as design a model discrimination algorithm to rule out models that are inconsistent with a newly observed output trajectory, by checking the feasibility of mixed-integer linear programs. Moreover, we investigate the trade-off among the accuracy of abstraction models, the computational cost for obtaining reduced models and the guaranteed detection time T for distinguishing the models. Finally, we evaluate the effectiveness of our approach on a vehicle intent estimation example using the highD data set of naturalistic vehicle trajectories recorded on German highways.**

I. INTRODUCTION

In most driving scenarios, model detection/identification plays an important role in safety validation and assurance such that potential changes in other vehicles' system behaviors can be quickly detected and dealt with. Since the precise model of other vehicles is often unknown, data-driven methods are widely used for analyzing such systems. For instance, Gaussian process regression was used to estimate driver actions and vehicle states in virtual car games [1], neural network was trained to fit the dynamical model of a scaled-down rally car [2] and a recurrent neural network was applied to deal with potential delays within vehicle models in an Apollo simulation system [3]. However, the performance of these methods in real vehicles has not been rigorously tested and most learning-based algorithms often do not analyze the modeling errors, which might lead to failure to satisfy safety requirements, e.g., collision avoidance. Further, the balance between model accuracy and computation cost should also be considered in real-time applications.

Literature Review. The model discrimination problem seeks to determine/identify which model in a *known* admissible model set can generate a finite sequence of measured input-output data [4], [5]. This can be reformulated as multiple model invalidation problems, which aim to check whether newly observed input-output data is compatible with one member in the model set [6]. When their mathematical models are given, several model invalidation algorithms are designed for linear parameter varying systems [7], [8], nonlinear systems [9], uncertain systems [10], switched auto-regressive models [11] and switched affine systems [5], [12], [13]. Moreover, the concept of T-distinguishability (or T-detectability) is introduced for analyzing the detectability of the models in [5], [13] to find a finite time horizon T within which a pair of models is guaranteed to be distinguished, if such a T exists. The notion of T-distinguishability is closely related to the concept of state/mode distinguishability of switched linear systems [14], [15], finite-state systems [16] and switched nonlinear systems [17].

However, exact mathematical models (for the admissible model set) are not always available in real-world applications. Thus, data-driven methods have been applied to identify or learn the unknown dynamics from data. The Gaussian process regression method, clustering based method and neural networks have been developed to predict future trajectories for the vehicle [18], [19] while fully-convolutional neural networks have been used to output both vehicle intent and corresponding intended trajectory [20], [21]. Moreover, when the unknown system dynamics are assumed to be Lipschitz or Hölder continuous, algorithms for approximating upper and lower bounding functions for the dynamics have been proposed in [22], [23]. These approaches belong to the class of non-parametric machine learning methods whose computation costs grow with the size of the data set.

Contribution. In this paper, we design a model discrimination algorithm for Lipschitz continuous systems whose exact models are not known. We first make use of a data-driven abstraction approach in [24] to obtain system models that over-approximate the unknown system dynamics. However, this model retains all the original data and thus, the use of this model will be computationally very expensive (when the data set is large). Hence, we propose a model reduction approach that obtains a piecewise affine abstraction with the number of pieces as a "tuning" parameter, which is a much more compact representation of the unknown dynamics and as such, algorithms based on this reduced model can scale better for high dimensional systems, while still framing/bracketing the original unknown dynamics.

Moreover, we propose an optimization method for analyzing the detectability of the reduced models via finding a finite time T within which the trajectories of any pair of models must differ at at least one time instance. Furthermore, with computation awareness in mind, we investigate the

M. Bhagwat, Z. Jin and S.Z. Yong are with School for Engineering of Matter, Transport and Energy, Arizona State Univ., Tempe, AZ 85287; (email: {mmbhagwa,zjin43,szyong}@asu.edu). This work was supported in part by DARPA grant D18AP00073 and NSF grant CNS-1943545. We acknowledge Research Computing at Arizona State University for providing High Performance Computing resources that have contributed to the results reported within this paper.

connection among the accuracy of the reduced abstraction models, the computation cost for synthesizing the reduced models and the number of time steps T required for T-distinguishablility by adjusting the number of pieces of the piecewise affine functions. Finally, we demonstrate the applicability of our approach for identifying vehicle intent by learning reduced models from the very large highD data set of naturalistic vehicle trajectories [25].

II. Preliminaries

A. Notation

For a vector $v \in \mathbb{R}^n$, $\|v\|_i$ denotes their i-norm with $i = \{1, 2, \infty\}$. For a set D, the size of the set is defined as $|D|$ while j-th element of the set is denoted as D^j.

B. Abstraction/Over-Approximation

The goal of an abstraction procedure is to over-approximate the original (possibly unknown) function $p = f(q) : \mathcal{Q} \subset \mathbb{R}^n \to \mathbb{R}$ by a pair of functions $\underline{f}$ and $\overline{f}$ (i.e., to find an abstraction model $\mathcal{H} \triangleq \{\overline{f}, \underline{f}\}$) such that the function $f(\cdot)$ is bounded/sandwiched by the pair of functions, i.e., $\underline{f}$ and $\overline{f}$ satisfy the following:

$$\underline{f}(q) \le f(q) \le \overline{f}(q), \ \forall q \in \mathcal{Q}. \tag{1}$$

1) Data Driven Abstraction: If only a noisy sampled data set $\mathcal{D} = \{(\tilde{q}_j, \tilde{p}_j)\}_{j=1,\dots,N}$ is available, where $\tilde{p}_j$ is the noisy measurement of $f(\tilde{q}_j)$, and the function $f(\cdot)$ is assumed to be Lipschitz continuous (i.e., there exists a positive finite-valued L_p, called the Lipschitz constant, such that $\forall q_1, q_2$ in domain of f, $|f(q_2) - f(q_1)| \le L_p \|q_2 - q_1\|_i$), a data abstraction can be found as follows.

Proposition 1. *[24] Consider a unknown Lipschitz function $f(\cdot)$ and its corresponding data set $\mathcal{D} = \{(\tilde{q}_j, \tilde{p}_j)\}_{j=1,\dots,N}$. For all $q \in \mathcal{Q}$, $\underline{f}_{\mathcal{D}}(\cdot)$ and $\overline{f}_{\mathcal{D}}(\cdot)$ are lower and upper abstraction functions for unknown function $f(\cdot)$, i.e., $\forall q \in \mathcal{Q}$, $\underline{f}_{\mathcal{D}}(q) \le f(q) \le \overline{f}_{\mathcal{D}}(q)$,*

$$\overline{f}_{\mathcal{D}}(q) = \min_{j \in \{1,\dots,N\}} (\tilde{p}_j + L_p \|q - \tilde{q}_j\|_i) + \varepsilon_t, \tag{2a}$$

$$\underline{f}_{\mathcal{D}}(q) = \max_{j \in \{1,\dots,N\}} (\tilde{p}_j - L_p \|q - \tilde{q}_j\|_i) - \varepsilon_t, \tag{2b}$$

with selected norm $i \in \{1, 2, \infty\}$ and $\varepsilon_t \triangleq \varepsilon_p + (L_p + 1)\varepsilon_q$, where ε_p and ε_q are noise bounds for $\tilde{p}_j$ and $\tilde{q}_j$, respectively. Moreover, $\underline{f}_{\mathcal{D}}$ and $\overline{f}_{\mathcal{D}}$ are also Lipschitz continuous functions with Lipschitz constant L_p.

Moreover, if the Lipschitz constant is unknown, the Lipschitz constant can be estimated from the data set $\mathcal{D}$ [24]:

$$\hat{L}_p = \max_{j \ne k} \frac{|\tilde{p}_j - \tilde{p}_k| - 2\varepsilon_p}{\|\tilde{q}_j - \tilde{q}_k\|_i + 2\varepsilon_q}. \tag{3}$$

2) Incremental Affine Abstraction: Further, to reduce/simplify a function/system, we will modify/extend the incremental affine abstraction approach in [26] to data-driven abstraction models (2) in Section IV-B, which relies on the following result to obtain an abstraction/over-approximation:

Proposition 2 ([27, Theorem 4.1 & Lemma 4.3])**.** *Let $\mathcal{Q}$ be an n-dimensional mesh element such that $\mathcal{Q} \subseteq \mathbb{R}^n$ with diameter δ (see definitions in [26]). Let $f : \mathcal{Q} \to \mathbb{R}$ be a Lipschitz continuous function with Lipschitz constant L_p and let f_l be the linear interpolation of $f(.)$ evaluated at the vertices of the mesh element S. Then, the approximation error σ defined as the maximum error between f and f_l on $\mathcal{Q}$, i.e., $\sigma = \max_{q \in \mathcal{Q}}(|f(q) - f_l(q)|)$, is upper-bounded by*

$$\sigma \le \sqrt{\tfrac{n}{2(n+1)}} L_p \delta.$$

C. Modeling Framework

Consider a constrained discrete-time nonlinear model $\mathcal{G}$:

$$\begin{aligned} \boldsymbol{x}_{t+1} &= f(\boldsymbol{x}_t, \boldsymbol{w}_t), \ \boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{v}_t, \\ \boldsymbol{x}_{t,i} &= f_i^c(\boldsymbol{z}_{t,i}), \ i = 1, \dots, n, \end{aligned} \tag{4}$$

where $\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^n$ is the state vector at time $t \in \mathbb{N}$, $\boldsymbol{u}_t \in \mathcal{U} \subset \mathbb{R}^m$ is a input vector, $\boldsymbol{y}_t \in \mathbb{R}^n$ is the measurement vector, $\boldsymbol{w}_t \in \mathcal{W} \subset \mathbb{R}^n$ is the bounded process noise, $\boldsymbol{v}_t \in \mathcal{V} \subset \mathbb{R}^m$ is the bounded measurement noise, $\boldsymbol{z}_{t,i} \triangleq \begin{bmatrix} x_{t,1}, \dots, x_{t,i-1}, x_{t,i+1}, \dots, x_{t,n} \end{bmatrix}^\top$ and f_i^c is a state-dependent function for state constraints.

To put this model in context, we consider the example of a vehicle model with the state vector defined as

$$\boldsymbol{x}_t \triangleq \begin{bmatrix} p_{x,t} & v_{x,t} & p_{y,t} & v_{y,t} \end{bmatrix}^\top, \tag{5}$$

where p_x and p_y are vehicle positions, while v_x and v_y are vehicle velocities along x- and y-axes, respectively. The vehicle dynamics is given by

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + [v_{x,t} \ a_{x,t} \ v_{y,t} \ v_{y,t}]^\top \Delta t + \boldsymbol{w}_t, \tag{6}$$

$$a_{x,t} = f^x(p_{x,t}, p_{y,t}, v_{x,t}, v_{y,t}), \tag{7}$$

$$a_{y,t} = f^y(p_{x,t}, p_{y,t}, v_{x,t}, v_{y,t}), \tag{8}$$

$$p_{y,t} = f^c(p_{x,t}, v_{x,t}, v_{y,t}), \tag{9}$$

with bounded process noise $\boldsymbol{w}_t$, unknown input function $f^x(\cdot)$, $f^y(\cdot)$ and state-dependent function $f^c(\cdot)$. We assume that the state of vehicle is fully observed with bounded measurement noise $\boldsymbol{v}_t$:

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{v}_t. \tag{10}$$

D. Model Discrimination

Finally, in preparation for solving the model discrimination problem, we adopt the definition in [5] of the length-T behavior of the original unknown model $\mathcal{G}$ and the abstracted model $\mathcal{H}$ based on the prior sampled data $\mathcal{D}$:

Definition 1 (Length-T Behavior of $\mathcal{G}$)**.** *The length-T behavior of the original (unknown) model $\mathcal{G}$ is the set of all length-T output trajectories that are compatible with $\mathcal{G}$, i.e.,*

$$\begin{aligned} \mathcal{B}_{t_0}^T(\mathcal{G}) := \{\{\boldsymbol{x}_t, \boldsymbol{y}_t\}_{t=t_0}^{t_0+T-1} \mid \exists \boldsymbol{x}_t \in \mathcal{X}, \boldsymbol{y}_t, \boldsymbol{w}_t \in \mathcal{W}, \\ \boldsymbol{v}_t \in \mathcal{V}, \ \textit{for } t \in \mathbb{Z}_{t_0+T-1}^{t_0}, \ \textit{s.t. } (4) \textit{ hold}\}. \end{aligned}$$

Definition 2 (Length-T Behavior of $\mathcal{H}$)**.** *The length-T behavior of the abstraction model $\mathcal{H}$ is the set of all length-T output trajectories that are compatible with $\mathcal{H}$, i.e.,*

$$\begin{aligned} \mathcal{B}_{t_0}^T(\mathcal{H}) := \{\{\boldsymbol{x}_t, \boldsymbol{y}_t\}_{t=t_0}^{t_0+T-1} \mid \exists \boldsymbol{y}_t \in \mathcal{Y}, \boldsymbol{w}_t \in \mathcal{W}, \\ \boldsymbol{v}_t \in \mathcal{V}, \ \textit{for } t \in \mathbb{Z}_{t_0+T-1}^{t_0}, \ \textit{s.t. } (1), (4) \textit{ hold}\}. \end{aligned}$$

Using the above definitions of system behaviors as well as the fact that $\mathcal{H}$ is an abstraction of $\mathcal{G}$ (by construction), we can conclude that $\mathcal{B}_{t_0}^T(\mathcal{G}) \subseteq \mathcal{B}_{t_0}^T(\mathcal{H})$.

III. Problem Statement

We now state the data-driven model discrimination problem that we consider in this paper. In particular, we want to determine how long it takes for models in a model set to be distinguished from each other given finite data:

Problem 1 (Detectability Analysis for a Set of Models $\{\mathcal{G}^l\}_{l=1}^{N_m}$)**.** *Given a set of constrained nonlinear models, $\{\mathcal{G}^l\}_{l=l}^{N_m}$, and a time horizon T, determine whether the set of models are T-distinguishable/detectable, i.e., whether $\exists t_0$ such that:*

$$\bigcap_{l=1}^{N_m} \mathcal{B}_{t_0}^T(\mathcal{G}^l) = \emptyset. \tag{11}$$

Problem 2 (Model Discrimination for $\{\mathcal{G}^l\}_{l=1}^{N_m}$)**.** *Given an output trajectory $\{\boldsymbol{y}_t\}_{t=t_0}^{t_0+T-1}$, a set of constrained nonlinear models $\{\mathcal{G}^l\}_{l=l}^{N_m}$ and a finite horizon T, determine which model the trajectory belongs to, i.e., find an i that for some t_0 satisfies*

$$\mathcal{B}_{t_0}^T(\mathcal{G}^i) \neq \emptyset \wedge (\mathcal{B}_{t_0}^T(\mathcal{G}^j) = \emptyset, \forall j \in \mathbb{Z}_1^{N_m}, j \neq i). \tag{12}$$

Since only a prior data set $\mathcal{D}_\mathcal{G}$ from the original model $\mathcal{G}$ with unknown dynamics is available, we propose to instead solve auxiliary problems with abstraction models, i.e., over-approximations of the models, which, if solved, also provide solutions to the above problems, i.e., in lieu of solving Problems 1 and 2, we will solve the following:

Problem 1.1 (Detectability Analysis for a Set of Abstraction Models $\{\mathcal{H}^l\}_{l=1}^{N_m}$)**.** *Given a set of abstraction models, $\{\mathcal{H}^l\}_{l=1}^{N_m}$ and a time horizon T, determine whether the set of models are T-distinguishable, i.e., whether $\exists t_0$ such that:*

$$\bigcap_{l=1}^{N_m} \mathcal{B}_{t_0}^T(\mathcal{H}^l) = \emptyset. \tag{13}$$

Problem 2.1 (Model Discrimination for $\{\mathcal{H}^l\}_{l=1}^{N_m}$)**.** *Given an output trajectory $\{\boldsymbol{y}_t\}_{t=t_0}^{t_0+T-1}$, a set of abstraction models $\{\mathcal{H}^l\}_{l=1}^{N_m}$, and a finite horizon T, determine which model the trajectory belongs to, i.e., find an i that for some t_0 satisfies*

$$\mathcal{B}_{t_0}^T(\mathcal{H}^i) \neq \emptyset \wedge (\mathcal{B}_{t_0}^T(\mathcal{H}^j) = \emptyset, \forall j \in \mathbb{Z}_1^{N_m}, j \neq i). \tag{14}$$

A good candidate for an abstraction model in the above auxiliary problems is the data-driven abstraction model $\mathcal{H}_\mathcal{D}$ obtained by applying Proposition 1. However, the computational time associated with this model will grow drastically with increasing size of the data set and state dimension. Therefore, we also propose to reduce its model complexity by addressing the following model reduction problem to obtain a reduced abstraction $\mathcal{H}$ for Problems 1.1 and 2.1:

Problem 3 (Model Reduction of Data-Driven Abstraction)**.** *For a set of N sampling data points $\mathcal{D}_\mathcal{G}$ and its corresponding data-driven abstraction $\mathcal{H}_\mathcal{D} \triangleq \{\overline{f}_\mathcal{D}, \underline{f}_\mathcal{D}\}$, find a pair of upper and lower piecewise affine functions $\overline{f}$ and $\underline{f}$ (i.e., $\mathcal{H} \triangleq \{\overline{f}, \underline{f}\}$) such that:*

$$\underline{f}(q) \leq \underline{f}_\mathcal{D}(q) \leq f(q) \leq \overline{f}_\mathcal{D}(q) \leq \overline{f}(q), \ \forall q \in \mathcal{Q}, \tag{15}$$

where $f(\cdot) : \mathcal{Q} \subset \mathbb{R}^n \to \mathbb{R}$ is the original unknown dynamics of the system, and $\overline{f}_\mathcal{D}, \underline{f}_\mathcal{D}$ are defined in Proposition 1.

Algorithm 1: Data Pre-Processing

Data: $\mathcal{D} = \{(\tilde{q}_\ell, \tilde{p}_\ell) | \ell = 1, \ldots, N\}$, δ_q
function dataProcess($\mathcal{D}$, δ_q)
  $\mathcal{D}_p = \{\}$;
  while $D \neq \emptyset$ **do**
    $\mathcal{D}_\mathcal{N} = \{\mathcal{D}^1\}$; $i = 2$;
    while $i \leq |D|$ **do**
      if $||\tilde{s}_i - \mu_{\mathcal{D}_\mathcal{N}}|| \leq \delta_x$ **then**
        $\mathcal{D}_\mathcal{N} \leftarrow \mathcal{D}^i$;
        Remove $\mathcal{D}^i$ from $\mathcal{D}$;
      else
        $i \leftarrow i + 1$
      end
    end
    Calculate a new point $(\mu_q, \mu_p, \epsilon_p, \epsilon_q)$ using (16);
    $\mathcal{D}_p \leftarrow (\mu_q, \mu_p, \epsilon_q, \epsilon_p)$;
  end
  return $\mathcal{D}_p$

Since the accuracy of this simplified model $\mathcal{H}$ is dependent on the number of pieces/subregions in the piecewise affine functions, we will further investigate the impact of varying the number of pieces on the computation time to find model $\mathcal{H}$ and the detection time T to distinguish all models.

IV. Main Approach

In this section, we first present a data pre-processing algorithm for naturalistic data and modify the incremental abstraction method to address Problem 3. Next, we adopt an optimization-based approach to solve Problem 1.1 (and, in turn, Problem 1) and Problem 2.1 (and, in turn, Problem 2).

A. Data Pre-Processing

Trajectories obtained from naturalistic data sets may have the same input $\tilde{q}_i = \tilde{q}_j$ with different outputs $\tilde{p}_i \neq \tilde{p}_j$. When directly applying Proposition 1 to these points, the procedure in (2) will choose the smaller $\tilde{p}$ for the upper function $\overline{f}(\cdot)$ and the larger $\tilde{p}$ for the lower function $\underline{f}(\cdot)$ so that $\underline{f}(\cdot)$ will be above the $\overline{f}(\cdot)$ when $s = \tilde{q}_i$ or $\tilde{q}_j$. However, this leads to a violation of the desired abstraction properties. To address this concern, we propose a data pre-processing method in Algorithm 1 such that the data-driven abstraction satisfies the desired abstraction properties in (2). We first select a constant δ_q and build a set $\mathcal{D}_\mathcal{N}$ consisting of adjacent data points. Instead of treating these data points separately, we replace these points with a single data point (μ_q, μ_p) with adjusted noise levels ϵ_q and ϵ_p using the following equations:

$$\begin{aligned} \mu_q &= \frac{\sum_{j=1}^{|\mathcal{D}_\mathcal{N}|} \tilde{q}_j}{|\mathcal{D}_\mathcal{N}|}, \ \epsilon_q = \max_{j \in \mathbb{Z}_1^{|\mathcal{D}_\mathcal{N}|}} \|\tilde{q}_j - \mu_q\|_i, \\ \mu_p &= \frac{\sum_{j=1}^{|\mathcal{D}_\mathcal{N}|} \tilde{p}_j}{|\mathcal{D}_\mathcal{N}|}, \ \epsilon_p = \max_{j \in \mathbb{Z}_1^{|\mathcal{D}_\mathcal{N}|}} |\tilde{p}_j - \mu_p|. \end{aligned} \tag{16}$$

B. Model Reduction

1) Incremental Affine Abstraction: To reduce the complexity of the data-driven abstraction model and overcome the limitations on space complexity for high dimensional systems, we now extend/modify the result in [26] to incrementally find a new abstraction model $\mathcal{H}$ to simplify the data-driven abstraction model $\mathcal{H}_\mathcal{D}$ using Algorithm 2 (see also

Algorithm 2: Procedures of Incremental Abstraction

1) Initialize $k=1$, $\mathcal{R}_0=\emptyset \implies \mathcal{V}_0=\emptyset$.
2) At increment k, consider a new sample set $\mathcal{S}'_k=(\mathcal{R}_k \setminus \mathcal{R}_{k-1}) \cup \mathcal{V}_{k-1}$ of size $\bar{c}$, where the set $(\mathcal{R}_k \setminus \mathcal{R}_{k-1}) \neq \emptyset$ denotes the newly added grid points such that $\mathcal{R}_k$ is expanding with k.
3) For the sample set $\mathcal{C}'_k$, use Lemma 1 to obtain hyperplanes $\mathcal{F}_k=\{\overline{f}_k, \underline{f}_k\}$ that over-approximate the data driven abstraction over $\mathcal{S}'_k$.
4) Go to step 2 with $k=k+1$ if $k<\kappa$.
5) After obtaining the final hyperplanes $\mathcal{F}_\kappa=\{\underline{f}_\kappa(q), \overline{f}_\kappa(q)\}$, the affine abstraction over the domain $\mathcal{Q}$ for the system (4) is:

$$\overline{f}(q)=\overline{E}_\kappa q+\overline{F}_\kappa+\sigma \mathbb{1}_n,$$
$$\underline{f}(q)=\underline{E}_\kappa q+\underline{F}_\kappa-\sigma \mathbb{1}_n,$$

where σ is the approximation error in Proposition 2.

the relevant definitions in [26]), to incrementally solve the abstraction problem, where a sequence of linear problems are solved with expanding operating regions with a subroutine given in the following lemma.

Lemma 1. *Given the affine abstraction model* $\mathcal{F}_{k-1}=\{\underline{f}_{k-1}(q), \overline{f}_{k-1}(q)\}$ *for the the data driven abstraction* $(\underline{f}_D(q), \overline{f}_D(q))$ *over an operating region* $\mathcal{R}_{k-1}$*, at increment* k*, solving the following problem over the sample set* $\mathcal{C}'_k=(\mathcal{R}_k \setminus \mathcal{R}_{k-1}) \cup \mathcal{V}_{k-1}$*, where* $\mathcal{V}_{k-1} \triangleq Ver(Conv(\mathcal{R}_{k-1}))$*, obtains an affine abstraction*[1] *of* $f(q)$ *over* $\mathcal{R}_k$*:*

$$\min_{\theta_k, \overline{E}_k, \underline{E}_k, \overline{F}_k, \underline{F}_k} \theta_k \tag{4}$$

subject to:

$\forall q \in \mathcal{R}_k \setminus \mathcal{R}_{k-1}:$

$$\begin{aligned} \overline{E}_k\, q+\overline{F}_k &\geq \overline{f}_D(q), \\ \underline{E}_k\, q+\underline{F}_k &\leq \underline{f}_D(q), \end{aligned} \tag{17a}$$

$\forall (q) \in \mathcal{V}_{k-1}:$

$$\begin{aligned} \overline{E}_k\, q+\overline{F}_k &\geq \overline{E}_{k-1}\, q+\overline{F}_{k-1}, \\ \underline{E}_k\, q+\underline{F}_k &\leq \underline{E}_{k-1}\, q+\underline{F}_{k-1}, \end{aligned} \tag{17b}$$

$\forall q \in \mathcal{V}_k=Ver(Conv(\mathcal{C}'_k)):$

$$(\overline{E}_k-\underline{E}_k)\, q+\overline{F}_k-\underline{F}_k \leq \theta_k \mathbb{1}_n, \tag{17c}$$

where the definitions of Ver *and* $Conv$ *and further details can be found in [26].*

Proof. The construction follows similar steps as in [26]. □

2) Partitions: Moreover, to further reduce the conservatism of the abstraction models, we extend our abstraction methods to consider multiple subregions/partitions, and the mesh-based abstraction is provided in Algorithm 3, with the abstraction method of a single subregion (see Algorithm 2) as the `abstraction` function. We first partition the given initial region (`bound`) into multiple subregions, where each subregion is hyperrectangular. For simplicity, in this paper, we present the function `divBound` to divide the state domain into a region set $\mathcal{I}=\{I_1,\dots,I_{m^n}\}$ by uniformly partitioning each interval $[\underline{q}_j, \overline{q}_j]$, $\forall j \in \mathbb{Z}_1^n$ into m subintervals of width $\frac{\overline{q}_j-\underline{q}_j}{m}$. Now, we have m^n subregions denoted by `subBounds` and the i-th subregion can be represented by $S_i q \leq \beta_i$. Then, the `abstraction` function can be carried out to obtain $\overline{f}_j$ and $\underline{f}_j$ for each subregion I_j.

[1]For brevity, we only provide the linear program for when the abstraction error is defined with $p=\infty$. Corresponding formulations for $p=\{1,2\}$ can be obtained with slight modifications, where we have a linear program when $p=1$, and a quadratically constrained linear program when $p=2$.

Algorithm 3: Abstraction for Multiple Subregions

Data: $\overline{f}_D$, $\underline{f}_D$, `bound` $=\mathcal{S}$, resolution r

```
function Partition(f̄_D, f_D, bound, r, ε_f)
    I ← divBounds(bound)
    for i = 1 : 2^n do
        cell{i} = abstraction(f̄_D, f_D, I_i, r)
    end
    return (cell, I)
function divBounds(bound)
    Refer to Section IV-B.1 for its description
    return (subBounds)
function abstraction(f̄_D, f_D, bound, r)
    Refer to Algorithm 2 for its description
    return (f̄, f)
```

3) Complexity Analysis: If the data set size is N_n and the computation cost for the norm $\|q-q_j\|_i$ is d, the computation cost to predict $\overline{f}(q)$ and $\underline{f}(q)$ for data-driven abstraction will be $\mathcal{O}(dN_n)$ as shown in [23], while the computation cost of our reduced model will be $\mathcal{O}(nM)$ if the number of subregions is M. Further, the space complexity will be reduced from $\mathcal{O}(nN_n)$ to $\mathcal{O}(nM)$. Since $M \ll N_n$ and $n \leq d$, both computation and space cost can be greatly saved.

C. Model Discrimination and T-Detectability

1) Abstraction Model: Armed with the abstraction tools described in Proposition 1 and Section IV-B, a "simpler" piecewise affine model $\mathcal{H}$ can be obtained from the data set $\mathcal{D}_\mathcal{G}$ of an unknown original model $\mathcal{G}$. Considering the vehicle model in (6) as an illustrative example, we can find the piecewise affine abstractions for $f^x(\cdot)$, $f^y(\cdot)$ and $f^c(\cdot)$:

$$\underline{E}_i^{f^x}\boldsymbol{x}+\underline{F}_i^{f^x} \leq f^x(\boldsymbol{x}) \leq \overline{E}_i^{f^x}\boldsymbol{x}+\overline{F}_i^{f^x}, \forall S_i\boldsymbol{x} \leq \beta_i, \tag{18a}$$

$$\underline{E}_i^{f^y}\boldsymbol{x}+\underline{F}_i^{f^y} \leq f^y(\boldsymbol{x}) \leq \overline{E}_i^{f^y}\boldsymbol{x}+\overline{F}_i^{f^y}, \forall S_i\boldsymbol{x} \leq \beta_i, \tag{18b}$$

$$\underline{E}_j^{f^c}\boldsymbol{z}+\underline{F}_j^{f^c} \leq f^c(\boldsymbol{z}) \leq \overline{E}_j^{f^c}\boldsymbol{z}+\overline{F}_j^{f^c}, \forall Q_j\boldsymbol{z} \leq \alpha_j, \tag{18c}$$

where $i \in \mathbb{Z}_1^{q_x}$, $j \in \mathbb{Z}_1^{q_z}$ and $\boldsymbol{z}=T_x^z\boldsymbol{x}$ while $T_x^z=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$. Note that the partitions of $f^x(\cdot)$ and $f^y(\cdot)$ can be different to achieve more accurate models but we assume them to be the same for simplicity in this paper.

Then, the state equation can be written as follows:

$$\underline{A}_i\boldsymbol{x}_t+\underline{h}_i+\underline{\boldsymbol{w}} \leq \boldsymbol{x}_{t+1} \leq \overline{A}_i\boldsymbol{x}_t+\overline{h}_i+\overline{\boldsymbol{w}}, \forall S_i\boldsymbol{x}_t \leq \beta_i, \tag{19}$$

with $\underline{A}_i=\begin{bmatrix}[1\ \ \Delta t\ \ 0\ \ 0]\\ [0\ \ 1\ \ 0\ \ 0]+\underline{E}_i^{f^x}\Delta t\\ [0\ \ 1\ \ 0\ \ \Delta t]\\ [0\ \ 1\ \ 0\ \ 0]+\underline{E}_i^{f^y}\Delta t\end{bmatrix}$ and $\underline{h}_i=\begin{bmatrix}0\\ \underline{F}_i^{f^x}\Delta t\\ 0\\ \underline{F}_i^{f^y}\Delta t\end{bmatrix}$, and $\overline{A}_i$ and $\overline{h}_i$ can be similarly defined with $\overline{E}_i^{f^x}$, $\overline{E}_i^{f^y}$, $\overline{F}_i^{f^x}$

and $\overline{F}_i^{f^y}$. Furthermore, the state-dependent constraint (9) can be equivalently written as:

$$\underline{E}_j^{f^c} \boldsymbol{z}_t + \underline{F}_j^{f^c} \leq T^z \boldsymbol{x} \leq \overline{E}_j^{f^c} \boldsymbol{z}_t + \overline{F}_j^{f^c}, \forall Q_j \boldsymbol{z}_t \leq \alpha_j, \quad (20)$$

where $\boldsymbol{z}_t = T_x^z \boldsymbol{x}_t$ and $T^z = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$. This equation can be further simplified to

$$B_j \boldsymbol{x}_t \leq b_j, \ \forall Q_j T_x^z \boldsymbol{x}_t \leq \alpha_j, \quad (21)$$

where $B_j = \begin{bmatrix} T^z - \overline{E}_j^{f^c} T_x^z \\ \underline{E}_j^{f^c} T_x^z - T^z \end{bmatrix}$ and $b_j = \begin{bmatrix} \overline{F}_j^{f^c} \\ -\underline{F}_j^{f^c} \end{bmatrix}$.

2) Detectability Analysis: Next, we present a T-distinguishability algorithm to find the guaranteed detection time T (i.e., to solve Problem 1.1) by solving the optimization problem below with increasing T:

Theorem 1 (T-Distinguishability/T-Detectability)**.** *A pair of constrained abstracted piecewise affine inclusion models $\mathcal{H}^i$ and $\mathcal{H}^j$, $i \neq j$ is T-distinguishable if the following is infeasible for any t_0 (with a search over t_0):*

Find $\boldsymbol{x}_t^\star, \boldsymbol{w}_t^\star, \boldsymbol{v}_t^\star, \boldsymbol{y}_t, a_{\sigma,t}^\star, s_{\sigma,t}^\star, \tilde{a}_{\tilde{\sigma},t}^\star, \tilde{s}_{\tilde{\sigma},t}^\star$
s.t. $\forall t \in \mathbb{Z}_{t_0+T-1}^{t_0}, \sigma \in \mathbb{Z}_1^{q_s}, \tilde{\sigma} \in \mathbb{Z}_1^{q_y}, \star \in \{i, j\}$:

$$\boldsymbol{x}_{t+1}^\star \leq \overline{A}_\sigma^\star \boldsymbol{x}_t^\star + \overline{h}_\sigma^\star + \boldsymbol{w}_t^\star + s_{\sigma,t}^\star \mathbb{1}_m, \quad (22a)$$
$$\boldsymbol{x}_{t+1}^\star \geq \underline{A}_\sigma^\star \boldsymbol{x}_t^\star + \underline{h}_\sigma^\star + \boldsymbol{w}_t^\star - s_{\sigma,t}^\star \mathbb{1}_m, \quad (22b)$$
$$S_\sigma^\star \boldsymbol{x}_t^\star \leq \beta_\sigma^\star + s_{\sigma,t}^\star \mathbb{1}_n, \quad (22c)$$
$$B_{\tilde{\sigma}}^\star \boldsymbol{x}_t^\star \leq b_{\tilde{\sigma}}^\star + \tilde{s}_{\tilde{\sigma},t}^\star \mathbb{1}_{(2n-2)}, \quad (22d)$$
$$Q_{\tilde{\sigma}}^\star T_x^z \boldsymbol{x}_t \leq \alpha_{\tilde{\sigma}}^\star + \tilde{s}_{\tilde{\sigma},t}^\star \mathbb{1}_{(n-1)}, \quad (22e)$$
$$\boldsymbol{y}_t = C \boldsymbol{x}_t + \boldsymbol{v}_t^\star, \quad (22f)$$
$$a_{\sigma,t}^\star \in \{0, 1\}, \ \tilde{a}_{\tilde{\sigma},t}^\star \in \{0, 1\}, \quad (22g)$$
$$SOS\text{-}1 : (a_{\sigma,t}^\star, s_{\sigma,t}^\star), \ SOS\text{-}1 : (\tilde{a}_{\tilde{\sigma},t}^\star, \tilde{s}_{\tilde{\sigma},t}^\star), \quad (22h)$$
$$\textstyle\sum_{\sigma=1}^{q_s} a_{\sigma,t}^\star = 1, \ \sum_{\tilde{\sigma}=1}^{q_y} \tilde{a}_{\tilde{\sigma},t}^\star = 1, \quad (22i)$$

where $s_{\sigma,t}^\star$ and $\tilde{s}_{\tilde{\sigma},t}^\star$ are slack variables, $s_{\sigma,t}^\star$ or $\tilde{s}_{\tilde{\sigma},t}^\star$ is free when $a_{\sigma,t}^\star$ or $\tilde{a}_{\tilde{\sigma},t}^\star$ is zero and zero otherwise (by virtue of the special ordered set of degree 1 (SOS-1) constraint).

Proof. $a_{\sigma,t}^\star = 1$ imply that (22a)–(22c) hold, since the SOS-1 constraints in (22h) ensure that $s_{\sigma,t}^\star = 0$. On the contrary, if $a_{\sigma,t}^\star = 0$, then $s_{\sigma,t}^\star$ is free and (22a)–(22c) hold trivially. Similarly, $\tilde{a}_{\tilde{\sigma},t}^\star = 1$, or $\tilde{a}_{\tilde{\sigma},t}^\star = 0$ implies that (22d)–(22e) hold. In addition, (22i) ensures that, at each time step t, only one partition is valid for each of the state and output equations. Therefore, if the above problem is infeasible, it means that there exists no common behavior that is satisfied by both models, i.e., $\mathcal{B}_{t_0}^T(\mathcal{H}^i) \bigcap \mathcal{B}_{t_0}^T(\mathcal{H}^j) = \emptyset$; hence, the pair of models is distinguishable from each other. ☐

3) Model Discrimination: Further, we present a model invalidation algorithm that enables us to discriminate among all models, i.e., to solve Problem 2.1, using Algorithm 4, if all model pairs are T-distinguishable according to Theorem 1. If not all pairs are T-distinguishable, Algorithm 4 will instead return the set of all models that are consistent with the given input-output data up to the current time step t_c.

Theorem 2. *Given a piecewise affine abstraction model $\mathcal{H}_\ell$ and a length-T input-output sequence $\{\boldsymbol{y}_t\}_{t=t_c-T+1}^{k=t_c}$ at time t_c, the model is invalidated if the following problem is infeasible:*

Find $\boldsymbol{x}_t, \boldsymbol{w}_t, \boldsymbol{v}_t, a_{i,t}, s_{i,t}, \tilde{a}_{j,t}, \tilde{s}_{j,t}$,
s.t. $\forall t \in \mathbb{Z}_{t_c}^{t_c-T+1}, \ i \in \mathbb{Z}_1^{q_s}, \ j \in \mathbb{Z}_1^{q_y}$:

$$\boldsymbol{x}_{t+1} \leq \overline{A}_i \boldsymbol{x}_t + \overline{h}_i + \boldsymbol{w}_t + s_{i,t} \mathbb{1}_n, \quad (23a)$$
$$\boldsymbol{x}_{t+1} \geq \underline{A}_i \boldsymbol{x}_t + \underline{h}_i + \boldsymbol{w}_t + s_{i,t} \mathbb{1}_m, \quad (23b)$$
$$S_i \boldsymbol{x}_t \leq \beta_i + s_{i,t} \mathbb{1}_n, \quad (23c)$$
$$a_{i,t} \in \{0, 1\}, \ SOS\text{-}1 : (a_{i,t}, s_{i,t}) \ \textstyle\sum_{i=1}^{n_s} a_{i,t} = 1, \quad (23d)$$
$$B_j \boldsymbol{x}_t \leq b_j + \tilde{s}_{j,t} \mathbb{1}_{(2n-2)}, \quad (23e)$$
$$Q_j T_x^z \boldsymbol{x}_t \leq \alpha_j + \tilde{z}_{j,t} \mathbb{1}_{(n-1)}, \quad (23f)$$
$$\tilde{a}_{j,t} \in \{0, 1\}, \ SOS\text{-}1 : (\tilde{a}_{j,t}, \tilde{s}_{j,t}) \ \textstyle\sum_{j=1}^{q_y} \tilde{a}_{j,t} = 1, \quad (23g)$$
$$\boldsymbol{y}_t = C \boldsymbol{x}_t + \boldsymbol{v}_t, \quad (23h)$$

where $s_{i,t}$ and $\tilde{s}_{j,t}$ are slack variables and $s_{i,t}$ or $\tilde{s}_{j,t}$ is free when $a_{i,t}$ or $\tilde{a}_{j,t}$ is zero and zero otherwise (by virtue of the special ordered set of degree 1 (SOS-1) constraint).

Proof. The construction follows similar steps to Theorem 1, but with only one model and with a given data sequence. ☐

Algorithm 4: Model Discrimination with Length T

```
Data: Models {G^l}_{l=1}^{N_m},
      Input-Output Sequence = {x_t, y_t}_{t=t_0-T+1}^{t_0}
function findModel({G^l}_{l=1}^{N_m}, {x_t, y_t}_{t=t_0-T+1}^{t_0})
    valid ← {G^l}_{l=1}^{N_m};
    for l = 1 : N_m do
        Check Feasibility of Theorem 2;
        if infeasible then
            Remove l from valid;
        end
    end
    return valid
```

V. CASE STUDY AND DISCUSSION

In this section, we demonstrate an implementation of the above defined algorithms for data-driven abstraction, model reduction, T-detectability and model invalidation using the highD data set consisting of naturalistic vehicle trajectories on German highways [25]. Specifically, we consider 3 distinct models, namely a collection of vehicles changing lanes in the left direction ('Left Turn'), changing lanes in the right direction ('Right Turn') and moving straight within the same lane ('Lane Keeping'). All simulations were done using MATLAB on a remote computing facility, Agave at Arizona State University, with the specification of 16 GB of memory.

A. HighD Data Set and Data Pre-Processing

In this section, we describe the highD data set and the data preparation/pre-processing we employed to simplify the use of various optimization algorithms defined in Section IV. The highD data includes detailed information about vehicles that pass over the designated highway over a certain period. The total amount of data available to use included trajectories from around $100,000$ vehicles out of which a subset of it was chosen to be the training set for obtaining the abstraction and a few other trajectories were extracted as the test set for implementing model invalidation. The data consists of the coordinates, along with the velocities and acceleration

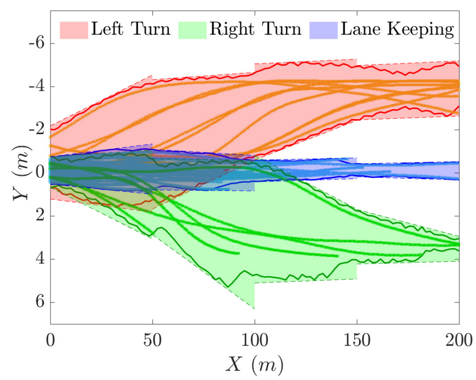


Fig. 1: Illustration of piecewise affine and data- driven abstractions of the unknown state-dependent function $p_y = f^c(p_x)$.

of the vehicles in the corresponding directions. In the data set, forward motion is chosen as the positive x direction and lane change is incorporated as lateral movement and a change in the y direction. The raw data is also normalized so that the vehicle trajectories always start from the origin. We use the vehicle model described in Section II-C and consider three vehicle intent models: Left Turn (Model I), Right Turn (Model II) and Lane Keeping (Model III).

B. Simulation Results

1) Abstraction Models: Figure 1 depicts the resulting abstraction models. Specifically, the 'kinky'/ragged upper and lower functions represent the data-driven abstraction via Proposition 1, while the shaded piecewise region is obtained by implementing the model reduction algorithm to the data-driven abstraction using Algorithm 2. The piecewise affine abstraction obtained is such that it encapsulates the 'kinky' data-driven abstraction and further helps in the simplification of the optimization problem for T-detectability and model invalidation. Both abstractions are further observed to encapsulate/over-approximate (some representative) actual data points on which these abstractions are based, as desired.

2) T-Detectability: The algorithm for T-detectability/T-distinguishability is implemented on the piecewise abstraction of a pair of models, from the 3 vehicle intentions. Varying numbers of subregions are considered for the piecewise abstraction and the corresponding detection times T and the CPU usage times are compared in Table I. From the table, it is evident that as the number of subregions increases, a larger number of optimization constraints have to be satisfied for each region and more binary variables are needed; as a result, the computation time clearly increases. Nonetheless, increasing the number of subregions is justified as it increases the closeness of the piecewise abstraction with the over approximated 'kinky' surface. However, the detection time T for all model pairs eventually plateaus. Consequently, increasing the number of regions is only justified up to a certain limit, beyond which the detection time would remain the same but the computation time is unnecessarily high.

3) Model Discrimination: Using the piecewise abstraction obtained, we implemented the model discrimination algorithm, where test trajectories were (in)validated for three different models that were learned. An illustration of the model discrimination results is shown in Figure 2, where three different test trajectories are (in)validated and the corresponding time step t is computed for when the trajectories are invalidated. The valid and invalid trajectories are represented by a flag, where 1 represents that the test trajectory belongs to the candidate model and a flag of 0 corresponds to that candidate model being invalidated for the trajectory.

TABLE I: Comparison of guaranteed detection time T for all model pairs, implemented for different numbers of subregions in the piecewise affine abstraction models.

	No. of regions, N	2	3	4
(i) Model I & Model II	T	167	165	171
	CPU Time (s)	94.55	409.69	1330.2
(ii) Model II & Model III	T	212	223	206
	CPU Time (s)	114.80	587.20	1513
(iii) Model I & Model III	T	172	179	167
	CPU Time (s)	101.54	403.86	1363.5

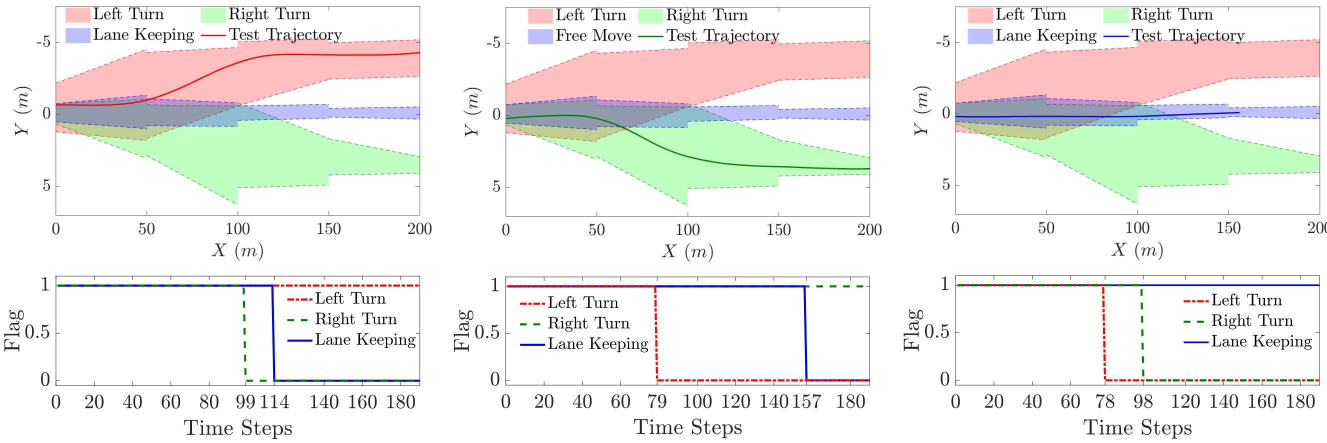
TABLE II: Comparison of model invalidation results for three different test trajectories, implemented for different numbers of subregions in the piecewise affine abstractions.

	No. of regions, N	2	3	4
(i) Model I	t_{max}	128	113	114
	CPU Time (s)	42.05	132.63	535.38
(ii) Model II	t_{max}	158	169	157
	CPU Time (s)	94.11	198.27	551.56
(iii) Model III	t_{max}	98	98	98
	CPU Time (s)	40.03	127.153	483.45

The model discrimination problem was then compared for different numbers of subregions in the model reduction algorithm. The values are compared in terms of the maximum time taken, t_{max} to invalidate a trajectory and the computation time taken to solve the optimization problem. The comparison is shown in Table II. Similar to the conclusions obtained from the comparison of T-detectability for the different number of partitions, it is also clear that the computation time increases with the increase in the number of partitions for model invalidation as well. The more the number of subregions/pieces chosen, the more accurate is the model, but after a certain number of pieces, the maximum steps required to invalidate the model plateaus. Thus, the computation time can be optimized by choosing the most appropriate number of subregions corresponding to the plateau value with the minimum CPU time.

VI. Conclusions

In this paper, we considered a model discrimination problem for Lipschitz continuous systems whose exact models are not known. We leveraged tools for incremental affine abstraction and data-driven abstraction to find simpler inclusion models for the unknown dynamics and proposed detectability analysis and model discrimination algorithms to estimate the true model using newly observed noisy data by checking the feasibility of the corresponding mixed-integer linear programs. Moreover, for the sake of computation



(a) Model I (b) Model II (c) Model III

Fig. 2: Sampled state trajectories (top row) and the corresponding model discrimination results (bottom row). In the state trajectory figure, solid lines denote the trajectories of three models and shadowed regions correspond to abstraction models with 4 regions. In the (bottom) figures depicting the model discrimination results, for $i \in \{\text{I, II, III}\}$, Flag i is 1 when the corresponding model i is validated and is 0, if invalidated. Model discrimination is achieved when only one Flag is 1.

awareness, we explored the trade-off among the accuracy of the abstraction models, the detectability time T and the computational cost in simulations. The effectiveness of our approach is illustrated on a vehicle intent estimation example using the highD data set consisting of naturalistic vehicle trajectories recorded on German highways.

References

[1] T. Georgiou and Y. Demiris, "Predicting car states through learned models of vehicle dynamics and user behaviours," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1240–1245.

[2] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *IEEE ICRA*, 2017, pp. 1714–1721.

[3] J. Xu, Q. Luo, K. Xu, X. Xiao, S. Yu, J. Hu, J. Miao, and J. Wang, "An automated learning-based procedure for large-scale vehicle dynamics modeling on baidu apollo platform," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 5049–5056.

[4] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[5] F. Harirchi, S. Z. Yong, and N. Ozay, "Guaranteed fault detection and isolation for switched affine models," in *IEEE Conference on Decision and Control (CDC)*, 2017, pp. 5161–5167.

[6] R. S. Smith and J. C. Doyle, "Model invalidation: A connection between robust control and identification," in *American Control Conference*. IEEE, 1989, pp. 1435–1440.

[7] F. D. Bianchi and R. S. Sánchez-Peña, "Robust identification/invalidation in an LPV framework," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 3, pp. 301–312, 2010.

[8] M. Sznaier and M. C. Mazzaro, "An LMI approach to control-oriented identification and model (in)validation of LPV systems," *IEEE Trans. on Automatic Control*, vol. 48, no. 9, pp. 1619–1624, 2003.

[9] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[10] Z. Jin, Q. Shen, and S. Z. Yong, "Optimization-based approaches for affine abstraction and model discrimination of uncertain nonlinear systems," in *IEEE Conf. on Decision and Contr.*,2019, pp. 7976–7981.

[11] N. Ozay, M. Sznaier, and C. Lagoa, "Convex certificates for model (in)validation of switched affine systems with unknown switches," *IEEE Trans. on Autom. Contr.*, vol. 59, no. 11, pp. 2921–2932, 2014.

[12] F. Harirchi and N. Ozay, "Guaranteed model-based fault detection in cyber–physical systems: A model invalidation approach," *Automatica*, vol. 93, pp. 476–488, 2018.

[13] F. Harirchi, S. Z. Yong, and N. Ozay, "Passive diagnosis of hidden-mode switched affine models with detection guarantees via model invalidation," in *Diagnosability, Security and Safety of Hybrid Dynamic and Cyber-Physical Systems*. Springer, 2018, pp. 227–251.

[14] H. Lou and R. Yang, "Conditions for distinguishability and observability of switched linear systems," *Nonlinear Analysis: Hybrid Systems*, vol. 5, pp. 427–445, 2011.

[15] M. Babaali and G. Pappas, "Observability of switched linear systems in continuous time," in *HSCC*, 2005, pp. 103–117.

[16] E. Santis and M. Benedetto, "Observability and diagnosability of finite state systems: A unifying framework," *Automatica*, vol. 81, pp. 115–122, 2017.

[17] N. Ramdani, L. Trave-Massuyes, and C. Jauberthie, "Mode discernibility and bounded-error state estimation for nonlinear hybrid systems," *Automatica*, vol. 91, pp. 118–125, 2018.

[18] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 141–146.

[19] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 591–594.

[20] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "TNT: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.

[21] S. Casas, W. Luo, and R. Urtasun, "IntentNet: Learning to predict intention from raw sensor data," in *Conference on Robot Learning*. PMLR, 2018, pp. 947–956.

[22] Z. Zabinsky, R. Smith, and B. Kristinsdottir, "Optimal estimation of univariate black-box Lipschitz functions with upper and lower error bounds," *Computers & Operations Research*, vol. 30, no. 10, pp. 1539–1553, 2003.

[23] J. Calliess, "Conservative decision-making and inference in uncertain dynamical systems," Ph.D. dissertation, University of Oxford, 2014.

[24] Z. Jin, M. Khajenejad, and S. Z. Yong, "Data-driven model invalidation for unknown lipschitz continuous systems via abstraction," in *2020 American Control Conference (ACC)*, 2020, pp. 2975–2980.

[25] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.

[26] S. M. Hassaan, M. Khajenejad, S. Jensen, Q. Shen, and S. Z. Yong, "Incremental affine abstraction of nonlinear systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 653–658, 2021.

[27] M. Stämpfle, "Optimal estimates for the linear interpolation error for simplices," *Journal of Approximation Theory*, vol. 103, pp. 78–90, 2000.