

# Data-Driven Model Discrimination of Switched Nonlinear Systems With Temporal Logic Inference

ZEYUAN JIN <sup>1</sup> (Student Member, IEEE), NASIM BAHARISANGARI <sup>1</sup>, ZHE XU <sup>1</sup> (Member, IEEE),  
AND SZE ZHENG YONG <sup>2</sup> (Member, IEEE)

(Formal Verification and Synthesis of Cyber-Physical Systems)

<sup>1</sup>School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287 USA

<sup>2</sup>Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115 USA

CORRESPONDING AUTHOR: SZE ZHENG YONG (e-mail: s.yong@northeastern.edu).

This work was supported in part by the NSF under Grants CNS-2312007, CNS-2313814, and CNS 2304863, and in part by ONR under Grant N00014-23-1-2505. (Zeyuan Jin and Nasim Baharisangari contributed equally to this work.)

**ABSTRACT** This article addresses the problem of data-driven model discrimination for unknown switched systems with unknown linear temporal logic (LTL) specifications, representing tasks, that govern their mode sequences, where only sampled data of the unknown dynamics and tasks are available. To tackle this problem, we propose data-driven methods to over-approximate the unknown dynamics and to infer the unknown specifications such that both set-membership models of the unknown dynamics and LTL formulas are guaranteed to include the ground truth model and specification/task. Moreover, we present an optimization-based algorithm for analyzing the distinguishability of a set of learned/inferred model-task pairs as well as a model discrimination algorithm for ruling out model-task pairs from this set that are inconsistent with new observations at run time. Further, we present an approach for reducing the size of inferred specifications to increase the computational efficiency of the model discrimination algorithms.

**INDEX TERMS** Fault detection and isolation, formal verification/synthesis, learning for control, model validation, nonlinear systems identification.

## I. INTRODUCTION

In cyber-physical systems (CPS) and multi-agent systems (MAS), various sensors, actuators, and subsystems/agents are interconnected, where each component in CPS or each agent in MAS may perform a distinct task while satisfying a distinct dynamical model. Distinguishing the components or agents and the tasks from each other becomes crucial for various purposes. For instance, to detect faults and minimize the risk of serious damages or failures in CPS, it is necessary to identify which component has abnormal functionality, while in MAS, it is desirable to detect the tasks/intents of other dynamic agents. For the purpose of distinguishing amongst different model behaviors, we can leverage the differences that are the result of different system dynamics or specifications, where the *system dynamics* governs the evolution of system states based on physical laws and the *specification* governs

the temporal evolution of their system modes corresponding to desired tasks or rules. The task specifications are often expressed using temporal logic formulas, such as linear temporal logic (LTL) formulas, which is a highly expressive language capable of providing formal yet easily understandable descriptions of system behaviors [1], [2]. Temporal logic formulas are widely used across many fields, including control synthesis and complex robotic applications, anomaly detection in underlying systems, and specification, recognition, and interpretation in dynamic environments [3], [4], [5], [6], [7], [8], [9], [10], [11].

Thus, discriminating among different models can be expedited by considering both the specifications and system dynamics in contrast to only utilizing the specifications or system dynamics alone. However, the exact specifications and system dynamics are usually not

available, which makes this problem more challenging and interesting.

**Related Work:** Model discrimination is the task of distinguishing between models based on a finite sequence of measured input-output data [12], [13], [14]. This problem can be approached by using a modeling invalidation framework, which seeks to determine whether the observed input-output data is consistent with any member of the set of valid models [15]. The problem of model invalidation has been studied for different types of systems, including linear parameter varying systems [16], [17], nonlinear systems [18], uncertain systems [19], switched auto-regressive models [20], and switched affine systems [13], [21], [22]. Additionally, to examine the detectability of the models,  $T$ -distinguishability (or  $T$ -detectability) is introduced in [13], [22] to find upper bounds on the required time horizon  $T$  to distinguish one model from the other, if such a  $T$  exists. The notion of  $T$ -distinguishability is closely related to the concept of state/mode distinguishability of switched linear systems [23], [24], finite-state systems [25] and switched nonlinear systems [26]. Recently, model discrimination and fault detection using temporal logics specifications have also gained attention. Jiang et al. [27] conducted failure diagnosis by specifying the fault as an LTL formula; and Yang et al. [28] combined switched affine systems and LTL formulas to constrain the switching modes so that the detectability horizon  $T$  can be reduced. Niu et al. [29] further extended the work in [28] to nonlinear switched systems, and they defined the specifications in the form of *metric/signal temporal logic* formulas. However, these approaches all rely on precise mathematical models of both the system dynamics and the temporal logic specifications.

When the system dynamics are unknown or only partially known, various learning approaches, e.g., Gaussian process regression, clustering-based methods and neural networks [30] have been developed. Further, to prevent the true model from being wrongly discriminated, set-membership learning approaches have also been proposed to over-approximate the unknown system dynamics from observed/sampled input-output data by finding a *set of known systems* that is guaranteed to contain the true model and retain most properties of interest with the unknown system dynamics [31], [32]. Within this context, the work in [33] proposes a recursive algorithm to compute upper and lower bounding functions for univariate Lipschitz continuous unknown dynamics while the study in [34], [35] considered the extension to multivariate functions. This technique was extended to unknown differentiable functions with bounded Jacobians in [36] and to Hölder continuous unknown dynamics in [37], while in [38], the authors further considered componentwise Hölder continuous functions where the contribution of each input to each output of the function is independently counted.

On the other hand, when the exact temporal logic formulas for the task specifications are unknown, the formulas can be inferred from traces of underlying systems and utilized for different purposes [6], [39], [40], [41], [42], [43], [44], [45], [46],

[47], [48], [49], [50], [51], [52], [53]. For example, LTL task specifications were inferred/learned from demonstrations of a task using Bayesian inference approach to address the problem of acceptability of task execution, as presented in [54]. Moreover, in [55], both formal grammar and temporal logic have been used for mining the structure and the parameters of a *signal temporal logic* specification from a set of unlabeled trajectories while in [56], [57], [58], formal grammar and temporal logic have been used for automated recognition of complex human activities and the task of run time verification. Nonetheless, the works mentioned above do not consider the learning and inference of the unknown system dynamics and task specifications simultaneously, which we find to be synergistic for the purpose of model discrimination.

**Contributions:** In this article, we aim to solve the problem of data-driven model discrimination among a set of unknown models with unknown switched system dynamics and unknown linear temporal logic (LTL) specifications that govern their mode sequences (representing tasks), where we only have access to data of the unknown dynamics and specifications. To address this challenge, we propose a data-driven set-membership method such that the learned dynamics model over-approximates the unknown dynamics and the ground truth LTL specification/task is included in the set of inferred specifications. Furthermore, we analyze the distinguishability of the learned data-driven model-task pairs by introducing an optimization-based algorithm that can find an upper bound on the required time horizon  $T$  to guarantee the discrimination/separation of the model-task pairs, as well as propose a model discrimination algorithm that can rule out learned model-task pairs that are inconsistent with new data/observations at run time. Moreover, we propose a complexity reduction framework for reducing the size of the inferred LTL formula by leveraging prior information, when available, that can increase the computation efficiency in model discrimination algorithms. Finally, we demonstrate the effectiveness of our proposed methods via illustrative examples of intent estimation for vehicles and single link robot arms. Our results indicate that the synergistic learning of both LTL specifications and system dynamics can significantly accelerate data-driven model discrimination.

## II. PRELIMINARIES

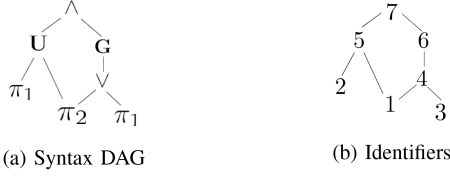
In this section, we introduce the mathematical concepts and notations used throughout this article.

### A. NOTATIONS

$\|v\|_p$  for  $p = \{1, \infty\}$  denotes the  $p$ -norm of a vector  $v \in \mathbb{R}^n$  and  $e_i$  is the unit basis vector with its  $i$ -th element being 1. The set of integers from  $a$  through  $b$  is denoted by  $\mathbb{Z}_a^b$ ,  $\mathbb{N} \triangleq \{1, 2, \dots\}$  and  $|S|$  is the cardinality of the finite set  $S$ .

### B. LINEAR TEMPORAL LOGIC (LTL)

First, we introduce an *atomic proposition* as a statement on a system variables denoted by  $\pi$ . This statement is either *True* or *False*. Let  $\Sigma$  be a finite set of atomic propositions. The



**FIGURE 1.** Syntax DAG and identifier of syntax DAG of the formula  $(\pi_1 \mathbf{U} \pi_2) \wedge \mathbf{G}(\pi_1 \vee \pi_2)$ .

syntax of LTL formulas over  $\Sigma$  is defined as follows.

$$\phi := \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathbf{X}\phi \mid \phi_1 \mathbf{U} \phi_2, \quad (1)$$

where  $\pi \in \Sigma$ ,  $\neg$  and  $\wedge$  are Boolean connectives,  $\mathbf{U}$  is the temporal operator “until”, and  $\mathbf{X}$  is the temporal operator “next”. We add syntactic sugar and define temporal operators  $\mathbf{G}$  (“always”) and  $\mathbf{F}$  (“finally”). In addition, we derive Boolean connectives  $\vee$  (“disjunction”) and  $\rightarrow$  (“implication”) from the introduced Boolean connectives. We denote the set containing all the temporal operators and all the Boolean connectives by  $\mathcal{O}$ .

Let  $\sigma$  be an  $\omega$ -word over  $\Sigma$  whose  $t$ -th element is denoted by  $\sigma_t$ . We define the Boolean semantics of LTL formulas over such sequences as follows.

$$\begin{aligned} (\sigma, t) &\models \pi \text{ iff } \sigma_t = \pi, \\ (\sigma, t) &\models \neg\phi \text{ iff } (\sigma, t) \not\models \phi, \\ (\sigma, t) &\models \phi_1 \wedge \phi_2 \text{ iff } (\sigma, t) \models \phi_1 \text{ and } (\sigma, t) \models \phi_2, \\ (\sigma, t) &\models \mathbf{X}\phi \text{ iff } (\sigma, t+1) \models \phi, \\ (\sigma, t) &\models \phi_1 \mathbf{U} \phi_2 \text{ iff } \exists t' \geq t : (\sigma, t') \models \phi_2 \\ &\text{and } \forall t'' \in [t, t') : (\sigma, t'') \models \phi_1. \end{aligned} \quad (2)$$

**Syntax DAG:** Any LTL formula can be represented as a syntax directed acyclic graph, i.e., syntax DAG. In a syntax DAG, the nodes are labeled with atomic propositions or temporal operators or Boolean connective that form an LTL formula [59]. For instance, Fig. 1(a) shows the unique syntax DAG of the formula  $(\pi_1 \mathbf{U} \pi_2) \wedge \mathbf{G}(\pi_1 \vee \pi_2)$ , in which the subformula  $\pi_2$  is shared. Fig. 1(b) shows the arrangement of the identifiers of each node in the syntax DAG. We denote the set of Boolean connectives and the temporal operators by  $\mathcal{C}$ . **Size of an LTL formula:** If we represent an LTL formula by a syntax DAG, then each node corresponds to a subformula; thus, the size of an LTL formula is the number of the DAG nodes. We denote the size of an LTL formula  $\phi$  by  $|\phi|$  [60].

### C. FORMAL GRAMMAR

**String:** If we denote an alphabet by  $\Pi$ , then a *string* over this alphabet is a finite sequence of symbols of  $\Pi$ .

**Kleene Closure:** Kleene closure, denoted by  $*$ , is a unary operation on sets of symbols. If we apply the Kleene operation on the alphabet  $\Pi$  (a finite set of symbols),  $\Pi^*$  is the set of all strings over  $\Pi$  including the empty string  $\epsilon$ .

**Formal Grammar:** A formal grammar is a set of productions (rules) that describe how to generate a string from the alphabet of a language. From a mathematical point of view, grammar is a quadruple  $(\Pi, \mathcal{N}, \mathcal{T}, \mathcal{R}\mathcal{U})$ . In this quadruple,  $\Pi$  is a finite non-empty set called *terminal alphabet*. The elements of this set are called *terminals* and are the symbols appearing in the output strings of a grammar. They are called terminals due to the fact that there is no rule in the grammar  $\mathcal{R}\mathcal{U}$  that changes them.  $\mathcal{N}$  is a finite nonempty set that is disjoint from  $\Pi$ . The elements of this set are called *non-terminals*. Non-terminals can be replaced by terminals.  $\mathcal{T} \in \mathcal{N}$  is a distinguished non-terminal which is called the “start” symbol.  $\mathcal{R}\mathcal{U}$  is a finite set of productions (rules) in the following form:  $\lambda \Rightarrow \mu$ , where  $\lambda \in (\Pi \cup \mathcal{N})^* \mathcal{N} (\Pi \cup \mathcal{N})^*$ , referred to as “head”, is a string of terminals and non-terminals, and contains at least one non-terminal symbol;  $\mu \in (\Pi \cup \mathcal{N})^*$ , referred to as “body”, is a string of terminals and non-terminals. The symbol  $\Rightarrow$  is commonly used in representing a production in formal grammar.

**Context-Free Grammar (CFG):** Context-free grammar (CFG) is a type of formal grammar in which the head consists of a single non-terminal symbol ( $\lambda \in \mathcal{N}$ ). CFGs are called context-free because any of the productions in  $\mathcal{R}\mathcal{U}$  can be applied to any of the non-terminals in  $\mathcal{N}$  regardless of whether the non-terminals are surrounded by other symbols or not. It is common to represent the productions that are applied to a same head in the same line [61], [62], [63]. The symbol  $\Rightarrow$  used in defining rules in formal grammar is different from the Boolean connective  $\rightarrow$  (“implies”) in LTL. The symbol  $\Rightarrow$  is used for defining a production between the head and the body for generating a string. In addition, the symbol  $|$  is commonly used for separating the “bodies” ( $\mu$ ) following a same “head” ( $\lambda$ ).

**Example 1:** We define  $\Pi = \{A, a, B, b, c, C\}$  and  $\mathcal{N} = \{\lambda_1, \lambda_2\}$ . We also have the following productions

$$\begin{aligned} \lambda_1 &\Rightarrow | \mu_1 | \mu_2, \\ \lambda_2 &\Rightarrow | \mu_3 | \mu_4, \end{aligned} \quad (3)$$

where  $\mu_1 := \lambda_2 a \lambda_2$ ,  $\mu_2 := \lambda_2 B \lambda_2$ ,  $\mu_3 := C c A$ , and  $\mu_4 := a B b$ . Using the productions in (3), we can generate the following strings:  $C c A a C c A$ ,  $a B b a a B b$ ,  $a B b a C c A$ ,  $a B b B a B b$ , etc. Note that in a CFG, we can also consider a single line of production or a subgroup of the productions to generate strings. For example, using the second line in (3), we generate  $C c a$ ,  $a B b$ .

### D. FORMULATION OF INTEGER ENCODING OF LTL FORMULAS

Next, we present the integer encoding of LTL formulas in [64], which will be used when solving the model discrimination problem in the Section VI.

The time evaluation of a given formula  $\phi$  is given by  $\mathbf{P}_\phi \triangleq \mathbf{P}_\phi^0 \mathbf{P}_\phi^1 \dots \mathbf{P}_\phi^T$ . For brevity, we will now only present constraints for the satisfaction of each operator of the LTL semantics, i.e.,  $(\sigma, t) \models \phi$  for the following operators, where  $\pi$ ,  $\pi'$  and  $\pi_i$  are

atomic propositions, and  $P_\phi^t$  is the truth value of formula  $\phi$  at time  $t$ , as defined in [64].

*Negation:* The formula  $\phi = \neg\pi$  can be modeled as:

$$P_\phi^t = (1 - P_\pi^t). \quad (4)$$

*Disjunction:* The formula  $\phi = \bigvee_{i=1}^k \pi_i$  can be modeled as:

$$P_\phi^t \leq \sum_{i=1}^k P_{\pi_i}^t; \quad P_\phi^t \geq P_{\pi_i}^t, i \in \mathbb{Z}_1^k. \quad (5)$$

*Conjunction:* The formula  $\phi = \bigwedge_{i=1}^k \pi_i$  can be modeled as:

$$P_\phi^t \geq \sum_{i=1}^k P_{\pi_i}^t - (k - 1); \quad P_\phi^t \leq P_{\pi_i}^t, i \in \mathbb{Z}_1^k. \quad (6)$$

*Next:* The formula  $\phi = \mathbf{X} p$  can be modeled as:

$$P_\phi^t = P_\pi^{t+1}. \quad (7)$$

*Until:* The formula  $\phi = \pi' \mathbf{U} \pi$  can be modeled as:

$$\begin{aligned} \alpha_{tj} &\geq P_{\pi'}^j + \sum_{\tau=t}^{j-1} P_{\pi'}^\tau - (j - t + 1), j \in \mathbb{Z}_{t+1}^T; \\ \alpha_{tj} &\leq P_{\pi'}^j, \alpha_{tj} \leq P_{\pi'}^\tau, j \in \mathbb{Z}_{t+1}^T, \tau \in \mathbb{Z}_t^j; \\ P_\phi^t &\leq \sum_{j=t}^T \alpha_{tj}; \quad P_\phi^t \geq \alpha_{tj}, j \in \mathbb{Z}_t^T. \end{aligned} \quad (8)$$

*Eventually:* The formula  $\phi = \mathbf{F} \pi$  can be modeled as:

$$P_\phi^t \leq \sum_{\tau=t}^T P_\pi^\tau; \quad P_\phi^t \geq P_\pi^\tau, \tau \in \mathbb{Z}_t^T. \quad (9)$$

*Always:* The formula  $\phi = \mathbf{G} \pi$  can be modeled as:

$$P_\phi^t \geq \sum_{\tau=t}^T P_\pi^\tau - (T - t); \quad P_\phi^t \leq P_\pi^\tau, \tau \in \mathbb{Z}_t^T. \quad (10)$$

### III. PROBLEM FORMULATION

#### A. MODELING FRAMEWORK

In this article, we consider two types of atomic propositions. One type of propositions represents modes of the system dynamics and the other type represents constraint modes indicating if certain state constraints are active or inactive; thus, we consider  $\Sigma = \Sigma_m \times \Sigma_s$  as a finite set of atomic propositions with  $\Sigma_m$  being the set of switched system modes and  $\Sigma_s$  being the set of state-dependent constraint modes. Specifically, we consider the  $N_p$  different unknown model-task pair  $(\{\mathcal{G}^l\}_{l=1}^{N_p}, \phi^l)$ , given by:

$$x_{t+1} = \begin{cases} f_{\sigma_m,1}^l(x_t, u_t) + w_t, & \text{if } \sigma_t^m = e_1, \\ \vdots \\ f_{\sigma_m,|\Sigma_m|}^l(x_t, u_t) + w_t, & \text{if } \sigma_t^m = e_{|\Sigma_m|}, \end{cases} \quad (11a)$$

$$x_{t,j}^c = g_{\sigma_s,j}^l(z_{t,j}), \text{ if } \sigma_t^{s,j} = 1, \forall j \in \mathbb{Z}_1^{|\Sigma_s|}, \quad (11b)$$

$$y_t = x_t + v_t, \quad (11c)$$

$$(\sigma, t) \models \phi^l, \quad (11d)$$

where  $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$  is the state vector at time  $t \in \mathbb{N}$ ,  $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$  is the control input vector,  $y_t \in \mathcal{Y} \subseteq \mathbb{R}^{n_y}$  is the observed output vector,  $w_t \in \mathcal{W} \subseteq \mathbb{R}^{n_w}$  is the process noise and  $v_t \in \mathcal{V} \subseteq \mathbb{R}^{n_v}$  is the measurement noise, where we assume that the sets  $\mathcal{X}$ ,  $\mathcal{U}$ ,  $\mathcal{W}$ ,  $\mathcal{Y}$ , and  $\mathcal{V}$  are known (bounded) polytopes.  $e_i$  is the unit basis vector,  $\sigma_t \triangleq \begin{bmatrix} \sigma_t^m \top & \sigma_t^s \top \end{bmatrix}^\top \in$

$\{0, 1\}^{|\Sigma_m|+|\Sigma_s|}$  and  $f_{\sigma_m,i}^l : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x}$  are functions describing the unknown state dynamics for each mode represented by atomic proposition  $\sigma_t^m = e_i$ , while  $g_{\sigma_s,j}^l : \mathcal{X}_j^c \rightarrow \mathcal{Z}_j$  are functions describing the unknown state-dependent modes for state (equality) constraints corresponding to each atomic proposition  $\sigma_t^{s,j} = 1$ . For each  $j$ ,  $\mathcal{X}_{t,j}^c$  and  $\mathcal{Z}_{t,j}$  are disjoint subsets/partitions of  $x_t$  (such that we do not have implicit equations in (11)) and correspondingly, their domains  $\mathcal{X}_j^c$  and  $\mathcal{Z}_j$  are also disjoint. For simplicity, we assume that the vector partition of  $x_t$  is given; however, if unknown, all combinations of  $x^c$  and  $z$  can be considered and any combination that lacks correlation (e.g., using statistical tests) can be disregarded. The trace  $\sigma$  satisfies an LTL formula  $\phi^l$  whose specification is unknown.

Moreover, we assume that the state dimension and number of modes are known and the switching of the modes occur (synchronously) at the discrete-time step of our observations, while the unknown functions are Lipschitz continuous.

*Assumption 1:* Each unknown function  $f(\cdot)$  (or  $g(\cdot)$ ), is  $L_p$ -Lipschitz continuous, i.e., there exists a positive finite-valued  $L_p > 0$ , called the Lipschitz constant, such that for all  $x_1, x_2$  in the domain of  $f$ ,  $|f(x_2) - f(x_1)| \leq L_p \|x_2 - x_1\|_p$  for  $p \in \{1, \infty\}$  and similarly, for all  $z_{1,i}, z_{2,i}$  in the domain of  $g$ ,  $|g(z_{2,i}) - g(z_{1,i})| \leq L_p \|z_{2,i} - z_{1,i}\|_p$  for  $p \in \{1, \infty\}$ .

In addition, we assume that we only have access to the data  $\mathcal{D}^l$  of (finite-length) input-mode-output trajectories for both positive and negative cases, where positive cases are collected when the system demonstrates the desired behavior satisfying the LTL formula, while negative cases are collected when the system demonstrates undesired behavior violating the LTL formula. Given a specific trajectory  $\{u_t, \sigma_t, y_t\}_{t=t_0}^{t_0+T-1}$  in this dataset,  $\{\sigma_t\}_{t=t_0}^{t_0+T-1}$  will be extracted as the traces data set  $\mathcal{D}_{LTL}^l$  while  $\{y_t, u_t, \sigma_t, y_{t+1}\}$  from  $t = t_0$  to  $t_0 + T - 2$  will be added to the input-output trajectories data set  $\mathcal{D}_\sigma^l$  according to  $\sigma_t$ . Consequently, the initial data set  $\mathcal{D}^l$  can be partitioned into a traces data set  $\mathcal{D}_{LTL}^l$  (consisting of positive and negative traces) and a input-output trajectories data set  $\mathcal{D}_\sigma^l$  for each  $\sigma \in \Sigma$ .

Further, in Section V-B, we also consider the special case when prior information about  $\phi$  is available to make the model discrimination more computationally efficient.

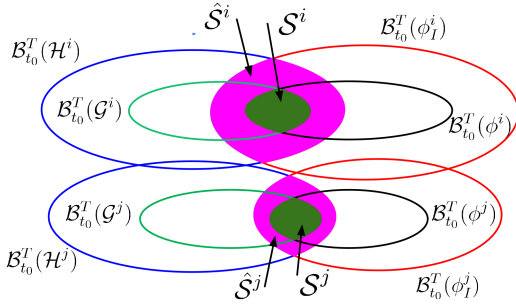
#### B. PROBLEM STATEMENT

To formulate the problems of interest, we adopt the following definitions.

*Definition 1:* We define a length- $T$  trajectory as  $\eta_T = \{u_t, \sigma_t, y_t\}_{t=t_0}^{t_0+T-1}$ , where  $u_t \in \mathcal{U}$  is the control input and  $\mathcal{U} \subseteq \mathbb{R}^m$ ,  $\sigma_t \in \Sigma$  is the mode at time  $t$  from the mode set  $\Sigma$ ,  $y_t$  is the system output, and  $T$  is a finite time horizon.

*Definition 2:* We define the length- $T$  behavior of a model  $\mathcal{G}^l$  at time  $t_0$  as the set of all length- $T$  trajectories  $\eta_T$  whose input-output sequences are compatible with  $\mathcal{G}^l$ , given by

$$\mathcal{B}_{t_0}^T(\mathcal{G}^l) := \{\{u_t, \sigma_t, y_t\}_{t=t_0}^{t_0+T-1} \mid \sigma_t \in \Sigma\},$$



**FIGURE 2.** For two model-task pairs  $(\mathcal{G}^i, \phi^i)$  and  $(\mathcal{G}^j, \phi^j)$  where the true system dynamics model and LTL formula of both pairs are unknown, we aim to find the  $\mathcal{H}$  and  $\phi_l$  from data so that the original system behavior is included by inferred system behavior ( $\mathcal{S} \subseteq \hat{\mathcal{S}}$ ). Thus, if the bigger magenta sets for model-task pairs  $i$  and  $j$  do not intersect, the two unknown model-task pairs are discriminated.

$$u_t \in \mathcal{U}, y_t \in \mathcal{Y}, x_t \in \mathcal{X},$$

$$w_t \in \mathcal{W}, v_t \in \mathcal{V}, \forall t \in \mathbb{Z}_{t_0}^{t_0+T-1} \text{ s.t. (11a)–(11c) holds}. \quad (12)$$

**Definition 3:** We define the length- $T$  behavior of an LTL formula  $\phi^l$  at time  $t_0$  as a set of all length- $T$  trajectories  $\eta_T$  whose mode sequences satisfy  $\phi^l$ , given by

$$\begin{aligned} \mathcal{B}_{t_0}^T(\phi^l) &:= \left\{ \{u_t, \sigma_t, y_t\}_{t=t_0}^{t_0+T-1} \mid \{\sigma_t\}_{t=t_0}^{t_0+T-1} \in V_{t_0}^T(\phi^l), \right. \\ &\quad \left. u_t \in \mathcal{U}, y_t \in \mathcal{Y}, \forall t \in \mathbb{Z}_{t_0}^{t_0+T-1} \right\}. \end{aligned} \quad (13)$$

Note in Definition 2, the mode sequences are not constrained by any specifications/LTL formulas. Further, the length- $T$  behaviors of the to-be-learned set-membership model  $\mathcal{H}^l$  and the to-be-inferred specification  $\phi_l^l$  can be similarly defined.

Having defined the above, we can now state our first problem of learning the unknown switched system dynamics  $\mathcal{G}^l$  and inferring the unknown LTL specification  $\phi^l$  from data with some specific inclusion properties:

**Problem 1 (Model Inference/Learning):** Given (prior) data sets  $\mathcal{D}_{LTL}^l$  and  $\mathcal{D}_\sigma^l$  for each  $\sigma \in \Sigma$  that are generated by the unknown model-task (i.e., dynamics-specification) pair  $(\mathcal{G}^l, \phi^l)$  for all  $l \in \mathbb{Z}_1^{N_p}$ , learn an over-approximation model  $\mathcal{H}^l$  of the original model  $\mathcal{G}^l$ , s.t.,  $\mathcal{B}_{t_0}^T(\mathcal{G}^l) \subseteq \mathcal{B}_{t_0}^T(\mathcal{H}^l)$  for all  $t_0$  and  $T$ , and infer the inferred specification  $\phi_l^l$  such that  $\phi^l \rightarrow \phi_l^l$  (equivalently,  $\mathcal{B}_{t_0}^T(\phi^l) \subseteq \mathcal{B}_{t_0}^T(\phi_l^l)$  for all  $t_0$  and  $T$ ).

Given the design that  $\mathcal{H}^l$  is an over-approximation of  $\mathcal{G}^l$  and  $\phi_l^l$  is implied by  $\phi^l$  as well as the definitions of system behaviors, we show that the intersection of the bigger circles (cf. Fig. 2) represent all possible learned models and inferred LTL specifications that are consistent with the data (including generalization errors) in following proposition:

**Proposition 1:** For the set  $\mathcal{S}^l = \mathcal{B}_{t_0}^T(\phi^l) \cap \mathcal{B}_{t_0}^T(\mathcal{G}^l)$  and the set  $\hat{\mathcal{S}}^l = \mathcal{B}_{t_0}^T(\phi_l^l) \cap \mathcal{B}_{t_0}^T(\mathcal{H}^l)$  containing the length- $T$  sequences generated by the true model-task pair  $(\mathcal{G}^l, \phi^l)$  and learned/inferred model-task pair  $(\mathcal{H}^l, \phi_l^l)$ , respectively,

if  $\mathcal{B}_{t_0}^T(\mathcal{G}^l) \subseteq \mathcal{B}_{t_0}^T(\mathcal{H}^l)$  and  $\phi^l \rightarrow \phi_l^l$  (i.e.,  $\mathcal{B}_{t_0}^T(\phi^l) \subseteq \mathcal{B}_{t_0}^T(\phi_l^l)$ ), then we have  $\mathcal{S}^l \subseteq \hat{\mathcal{S}}^l$ .

*Proof:* We assume that  $\eta \in \mathcal{S}^l = \mathcal{B}_{t_0}^T(\phi^l) \cap \mathcal{B}_{t_0}^T(\mathcal{G}^l)$  is an arbitrary length- $T$  sequence generated by true model-task pair  $(\mathcal{G}^l, \phi^l)$ : 1) for any  $\eta \in \mathcal{S}^l$ , we have  $(\eta, t_0) \models \phi^l$ ; thus, if  $\phi^l \rightarrow \phi_l^l$ , then we conclude that for any  $\eta \in \mathcal{S}^l$  we have  $(\eta, t_0) \models \phi_l^l$  (i.e., for any  $\eta \in \mathcal{S}^l$ , we have  $\eta \in \mathcal{B}_{t_0}^T(\phi_l^l)$ ); 2) we know that  $\eta \in \mathcal{B}_{t_0}^T(\mathcal{G}^l)$  and  $\mathcal{B}_{t_0}^T(\mathcal{G}^l) \subseteq \mathcal{B}_{t_0}^T(\mathcal{H}^l)$ ; therefore, for any  $\eta \in \mathcal{S}^l$ , we have  $\eta \in \mathcal{B}_{t_0}^T(\mathcal{H}^l)$ . By considering 1) and 2), we conclude that  $\mathcal{S} \subseteq \hat{\mathcal{S}}$ , where  $\hat{\mathcal{S}}^l = \mathcal{B}_{t_0}^T(\phi_l^l) \cap \mathcal{B}_{t_0}^T(\mathcal{H}^l)$ . ■

Then, if the bigger magenta sets for model-task pairs  $i$  and  $j$  do not intersect (cf. Fig. 2), i.e.,  $\hat{\mathcal{S}}^i \cap \hat{\mathcal{S}}^j = \emptyset$ , it is straightforward to conclude that the behaviors of original model-task pair (green sets) are also disjoint:

**Proposition 2:** For the true model-task pairs  $(\mathcal{G}^i, \phi^i)$  and  $(\mathcal{G}^j, \phi^j)$ , and the learned/inferred model-task pairs  $(\mathcal{H}^i, \phi_l^i)$  and  $(\mathcal{H}^j, \phi_l^j)$ , we have  $\hat{\mathcal{S}}^i \cap \hat{\mathcal{S}}^j = \emptyset \rightarrow \mathcal{S}^i \cap \mathcal{S}^j = \emptyset$ , where  $\hat{\mathcal{S}}^i = \mathcal{B}_{t_0}^T(\phi_l^i) \cap \mathcal{B}_{t_0}^T(\mathcal{H}^i)$ ,  $\hat{\mathcal{S}}^j = \mathcal{B}_{t_0}^T(\phi_l^j) \cap \mathcal{B}_{t_0}^T(\mathcal{H}^j)$ ,  $\mathcal{S}^i = \mathcal{B}_{t_0}^T(\phi^i) \cap \mathcal{B}_{t_0}^T(\mathcal{G}^i)$ , and  $\mathcal{S}^j = \mathcal{B}_{t_0}^T(\phi^j) \cap \mathcal{B}_{t_0}^T(\mathcal{G}^j)$ .

Equipped by Propositions 1 and 2, we now state the following data-driven model discrimination problems to determine how long it takes for the model-task pairs in a model-task pair set to be distinguished from each other by way of the distinction of their learned/inferred model-task pairs:

**Problem 2:** (Distinguishability for a set of true model-task pairs). Given a set of learned/inferred model-task pairs  $\{(\mathcal{H}^l, \phi_l^l)\}_{l=1}^{N_p}$  and a finite time horizon  $T$ , determine whether the set of learned/inferred model-task pairs is T-distinguishable, i.e., whether  $\exists t_0$  such that  $\bigcap_{l=1}^{N_p} (\mathcal{B}_{t_0}^T(\mathcal{H}^l) \cap \mathcal{B}_{t_0}^T(\phi_l^l)) = \emptyset$ .

**Problem 3:** (Model-task pair discrimination) Given a (newly observed) length- $T$  input-mode-output sequence  $\{u_t, \sigma_t, y_t\}_{t=t_c-T+1}^{t_c}$  at the current time  $t_c$  and a set of learned/inferred model-task pairs  $\{(\mathcal{H}^l, \phi_l^l)\}_{l=1}^{N_p}$ , determine which learned/inferred model-task pair it is based on the given sequence. That is to find an  $i \in \mathbb{Z}_1^{N_p} = \{1, 2, \dots, N_p\}$  that satisfies  $(\mathcal{B}_{t_0}^T(\mathcal{H}^i) \cap \mathcal{B}_{t_0}^T(\phi_l^i) \neq \emptyset) \wedge (\mathcal{B}_{t_0}^T(\mathcal{H}^j) \cap \mathcal{B}_{t_0}^T(\phi_l^j) = \emptyset)$ ,  $\forall j \in \mathbb{Z}_1^{N_p}$  and  $i \neq j$  with  $t_0 = t_c - T + 1$ .

Our approach to solve Problem 1 consists of a few steps. We first divide the data into trajectory and trace data, and then, we infer the system dynamics and LTL specifications separately in Sections IV and V, respectively. In addition, when prior information about the family of possible LTL specifications is known or given, we also consider formula reduction techniques in Sections V-B. Finally, using the learned system dynamics and inferred LTL formulas (from Problem 1), we address Problems 2 and 3 in Section VI.

#### IV. SYSTEM DYNAMICS LEARNING

In this section, we learn over-approximations of the unknown true system dynamics from the trajectory data  $\mathcal{D}_\pi^l$  for all

$\pi \in \Sigma$  using set-membership methods in [35], [65], [66]. To illustrate the proposed over-approximation method, we consider a generic original unknown function  $r_o = f_a(r) : \mathbb{R} \subset \mathbb{R}^{n_r} \rightarrow \mathbb{R}$  representing  $f_{\sigma^m,i}^l$  or  $g_{\sigma^s,j}^l$ . The goal of an over-approximation procedure is to find a pair of functions  $\underline{f}_a$  and  $\bar{f}_a$  (i.e., to find an over-approximation model  $\mathcal{H} \triangleq \{\bar{f}_a, \underline{f}_a\}$ ) such that the function  $f_a(\cdot)$  is bounded by the pair of functions, i.e.,  $\underline{f}_a$  and  $\bar{f}_a$  satisfy the following:

$$\underline{f}_a(r) \leq f_a(r) \leq \bar{f}_a(r), \quad \forall r \in \mathbb{R}. \quad (14)$$

If only a noisy sampled input-output data set  $\mathcal{D}_r = \{(\tilde{r}_j, \tilde{r}_{o,j}) | j = 1, \dots, N_D\}$  is available and function  $f_a(\cdot)$  is assumed to be Lipschitz continuous, a data abstraction can be found by following proposition according to [35].

**Proposition 3:** Consider a unknown Lipschitz function  $f_a(\cdot)$  and its corresponding noisy data set  $\mathcal{D}_r = \{(\tilde{r}_j, \tilde{r}_{o,j}) | j = 1, \dots, N_D\}$ . For all  $r \in \mathbb{R}$ ,  $\underline{f}_{a,D}(\cdot)$  and  $\bar{f}_{a,D}(\cdot)$  are lower and upper abstraction functions for unknown function  $f_a(\cdot)$ , i.e.,  $\forall r \in \mathbb{R}, \underline{f}_{a,D}(r) \leq f_a(r) \leq \bar{f}_{a,D}(r)$ ,

$$\bar{f}_{a,D}(r) = \min_{j \in \{1, \dots, N_D\}} (\tilde{r}_{o,j} + L_p \|r - \tilde{r}_j\|_p) + \varepsilon_t, \quad (15a)$$

$$\underline{f}_{a,D}(r) = \max_{j \in \{1, \dots, N_D\}} (\tilde{r}_{o,j} - L_p \|r - \tilde{r}_j\|_p) - \varepsilon_t, \quad (15b)$$

with a selected norm  $p \in \{0, 1, \infty\}$  and  $\varepsilon_t \triangleq \varepsilon_{r_o} + (L_p + 1)\varepsilon_r$ , where  $\varepsilon_r \triangleq \max(\tilde{r} - r)$  and  $\varepsilon_{r_o} \triangleq \max(\tilde{r}_o - r_o)$  are bounds on the measurement noise/errors for  $r$  and  $r_o$ , respectively.

When the Lipschitz constant is unknown, the Lipschitz constant can also be estimated from the dataset  $\mathcal{D}_r = \{(\tilde{r}_j, \tilde{r}_{o,j}) | j = 1, \dots, N_D\}$  as follows, according to [35]:

$$\hat{L}_p^{f_a} = \max_{j \neq i} \frac{|\tilde{r}_{o,j} - \tilde{r}_{o,i}| - 2\varepsilon_{r_o}}{\|\tilde{r}_j - \tilde{r}_i\|_p + 2\varepsilon_r}. \quad (16)$$

**Remark 1:** A classical result on *probably approximately correct* (PAC) [67] learning can be leveraged to relate confidence levels for Lipschitz constant estimation with dataset size [35]; thus, we can estimate the Lipschitz constants with high probability/confidence given sufficient data.

The method can be applied to find an inclusion model that over-approximates the unknown system dynamics and we can achieve more precise estimation of the Lipschitz constants and a more accurate inclusion model with an increase of the dataset size. However, this data-driven approach belongs to the class of non-parametric learning methods in the machine learning literature that are known to not scale well with the size of the datasets. Thus, to further simplify the data-driven model in  $\mathcal{H}^l$ , i.e., the functions  $f_{\sigma^m}^l$  and  $g_{\sigma^s,j}^l$  for each  $\sigma^m \in \{e_1, \dots, e_{|\Sigma_m|}\}$  and each  $\sigma^s \in \{0, 1\}^{|\Sigma_s|}$ , we propose to leverage a result in [66], [68] to over-approximate the nonlinearities with piecewise affine inclusions. First, we introduce the definition of the partitions.

**Definition 4 (Partition):** For each function  $f_{\sigma^m}^l$  or  $g_{\sigma^s,j}^l$ , a partition  $\mathcal{I}_{\sigma^m}^{f,l}$  of the closed bounded region  $\mathcal{X} \times \mathcal{U} \subseteq \mathbb{R}^{n_x+n_u}$

is a collection of  $q_{\sigma^m}^{f,l}$  subregions  $\mathcal{I}_{\sigma^m}^{f,l} = \{I_{\sigma^m,\star}^{f,l} | \star \in \mathbb{Z}_1^{q_{\sigma^m}^{f,l}}\}$  such that  $\mathcal{X} \times \mathcal{U} \subseteq \bigcup_{\star=1}^{q_{\sigma^m}^{f,l}} I_{\sigma^m,\star}^{f,l}$  and  $I_{\sigma^m,\star}^{f,l} \cap I_{\sigma^m,\star'}^{f,l} = \partial I_{\sigma^m,\star}^{f,l} \cap \partial I_{\sigma^m,\star'}^{f,l}, \forall \star \neq \star' \in \mathbb{Z}_1^q$ , where  $\partial I_{\sigma^m,\star}^{f,l}$  is the boundary of set  $I_{\sigma^m,\star}^{f,l}$ . Similarly, a partition  $\mathcal{I}_{\sigma^s,j}^{g,l} = \{I_{\sigma^s,j,\dagger}^{g,l} | \dagger \in \mathbb{Z}_1^{q_{\sigma^s,j}^{g,l}}\}$  with  $q_{\sigma^s,j}^{g,l}$  subregions of the closed bounded region  $\mathcal{Z}_j$  for vector fields  $g_{\sigma^s,j}^l$  can be defined.

We assume that the partitions are polytopic. Then, for each polytopic subregion  $I_{\sigma^m,\star}^{f,l} \in \mathcal{I}_{\sigma^m}^{f,l}$  (or  $I_{\sigma^s,j,\dagger}^{g,l} \in \mathcal{I}_{\sigma^s,j}^{g,l}$ ) that partitions the domain of interest, the unknown function  $f_{\sigma^m}^l$  (or  $g_{\sigma^s,j}^l$ ) can be over-approximated/abstracted by a pair of affine functions  $\underline{f}_{\sigma^m,\star}^l, \bar{f}_{\sigma^m,\star}^l$  (or  $\underline{g}_{\sigma^s,j,\dagger}^l, \bar{g}_{\sigma^s,j,\dagger}^l$ ) by solving a linear programming (LP) problem similar to [66], [68] that minimizes the maximum distance between the upper and lower affine functions/hyperplanes for each partition. As a result, for all  $(x, u) \in I_{\sigma^m,\star}^{f,l}$  (or  $z_j \in I_{\sigma^s,j,\dagger}^{g,l}$ ), the function  $f_{\sigma^m}^l(x, u)$  (or  $g_{\sigma^s,j}^l(z_j)$ ) is sandwiched/framed by a pair of affine functions, i.e.,  $\underline{f}_{\sigma^m}^l(x, u) \leq \underline{f}_{D,\sigma^m}^l(x, u) \leq \bar{f}_{D,\sigma^m}^l(x, u) \leq \bar{f}_{\sigma^m}^l(x, u)$  (or  $\underline{g}_{\sigma^s,j}^l(z_j) \leq \underline{g}_{D,\sigma^s,j}^l(z_j) \leq \bar{g}_{D,\sigma^s,j}^l(z_j) \leq \bar{g}_{\sigma^s,j}^l(z_j)$ ) with

$$\begin{aligned} \underline{f}_{\sigma^m,\star}^l(x, u) &= \underline{A}_{\sigma^m,\star}^l x + \underline{B}_{\sigma^m,\star}^l u + \underline{h}_{\sigma^m,\star}^l, \\ \bar{f}_{\sigma^m,\star}^l(x, u) &= \bar{A}_{\sigma^m,\star}^l x + \bar{B}_{\sigma^m,\star}^l u + \bar{h}_{\sigma^m,\star}^l, \\ \underline{g}_{\sigma^s,j,\dagger}^l(z_j) &= \underline{H}_{\sigma^s,j,\dagger}^l z_j + \underline{h}_{\sigma^s,j,\dagger}^l, \\ \bar{g}_{\sigma^s,j,\dagger}^l(z_j) &= \bar{H}_{\sigma^s,j,\dagger}^l z_j + \bar{h}_{\sigma^s,j,\dagger}^l, \end{aligned} \quad (17)$$

where  $\underline{A}_{\sigma^m,\star}^l, \bar{A}_{\sigma^m,\star}^l, \underline{B}_{\sigma^m,\star}^l, \bar{B}_{\sigma^m,\star}^l, \underline{H}_{\sigma^s,j,\dagger}^l, \bar{H}_{\sigma^s,j,\dagger}^l, \underline{h}_{\sigma^m,\star}^l, \bar{h}_{\sigma^m,\star}^l, \underline{h}_{\sigma^s,j,\dagger}^l$  and  $\bar{h}_{\sigma^s,j,\dagger}^l$  are of appropriate dimensions and are constants that are determined by the piecewise affine abstraction algorithm in [66], [68].

Since the original functions  $f(\cdot)$  and  $g(\cdot)$  are bounded by data-driven over-approximation  $\bar{f}_D(\cdot)$  and  $\underline{f}_D(\cdot)$ , the following piecewise affine interval models  $\mathcal{H}^l$  is an over-approximation for the original dynamics model  $\mathcal{G}^l$  satisfying  $\mathcal{B}_{t_0}^T(\mathcal{G}^l) \subseteq \mathcal{B}_{t_0}^T(\mathcal{H}^l)$  for all  $t_0$  and  $T$ :

$$\begin{pmatrix} \underline{A}_{\sigma_t^m,\star}^l x_t + \underline{B}_{\sigma_t^m,\star}^l u_t \\ + \underline{h}_{\sigma_t^m,\star}^{f,l} \end{pmatrix} \leq x_{t+1} \leq \begin{pmatrix} \bar{A}_{\sigma_t^m,\star}^l x_t + \bar{B}_{\sigma_t^m,\star}^l u_t \\ + \bar{h}_{\sigma_t^m,\star}^{f,l} \end{pmatrix},$$

$$y_t = Cx_t + v_t,$$

$$\underline{H}_{\sigma_t^s,j,\dagger}^l z_{t,j} + \underline{h}_{\sigma_t^s,j,\dagger}^{g,l} \leq x_{t,j}^c \leq \bar{H}_{\sigma_t^s,j,\dagger}^l z_{t,j} + \bar{h}_{\sigma_t^s,j,\dagger}^{g,l}, \quad (18)$$

where their corresponding polytopic subregions  $I_{\sigma_t^m,\star}^{f,l}$  and  $I_{\sigma_t^s,j,\dagger}^{g,l}$  can be represented by the following linear constraints:

$$S_{\sigma_t^m,\star}^{x,l} x_t + S_{\sigma_t^m,\star}^{u,l} u_t \leq \beta_{\sigma_t^m,\star}^l,$$

**Algorithm 1:** Inference of LTL Formulas.

---

**Input:**  $\mathcal{D}_{LTL}$ ; Maximum formula size  $Itr \in \mathbb{N}$

```

1  $E_\gamma = E_{LTL} = \emptyset$ 
2  $i \leftarrow 1$ 
3  $k \leftarrow 1$ 
4 while  $k \leq Itr$  do
5   Construct  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  of size  $k$  consistent with  $\mathcal{D}_{LTL}$ 
6   if  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  is satisfiable then
7     Compute a model  $\gamma$  of  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$ 
8     Store  $\gamma$  in  $E_\gamma$ 
9     Add  $\bigwedge_{k'=1, \dots, |E_\gamma|} \bigvee_{\rho \in Var_{\gamma_{k'}}^1} (\rho \notin Var_{\gamma_{k'}}^1)$  to the
       constraints  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$ 
10     $i \leftarrow i + 1$ 
11  else
12     $E_\gamma = \emptyset$ 
13     $k \leftarrow k + 1$ 
14  end if
15 end while
16  $N_o \leftarrow i - 1$ 
17  $\phi_I \leftarrow \bigvee_{i=1, \dots, |E_{LTL}|} \phi^i$ 
18 return  $\phi_I$ 

```

---

$$S_{\sigma_i^{s,j}, \dagger}^{z,l} z_i \leq \beta_{\sigma_i^{s,j}, \dagger}^l \quad (19)$$

respectively, with  $S_{\sigma_i^{m,*}}^{x,l}$ ,  $S_{\sigma_i^{m,*}}^{u,l}$ ,  $S_{\sigma_i^{s,j}, \dagger}^{z,l}$ ,  $\beta_{\sigma_i^{m,*}}^l$  and  $\beta_{\sigma_i^{s,j}, \dagger}^l$  of appropriate dimensions.

Note that the precision of the inferred/learned model can be improved with more and better chosen *partitions*, but may result in longer computation times since more integer variables will be introduced in our solutions in Section VI.

## V. LTL SPECIFICATION INFERENCE

We first propose an algorithm for learning the inferred specification  $\phi_I$  (without any prior information). Then, we propose a framework for reducing the size of the inferred formula, when prior information is available, for further improving the computational efficiency of the model discrimination algorithms in Section VI.

### A. LEARNING THE UNKNOWN LTL SPECIFICATION

In this subsection, we present an algorithm for learning  $\phi_I$  from a given dataset of finite traces  $\mathcal{D}_{LTL}$ .  $\mathcal{D}_{LTL}$  consists of two subsets of traces: 1) the subset of *positive traces* denoted by  $\mathcal{D}_{LTL}^{pos}$  (satisfying the true LTL formula  $\phi$ ) and 2) the subset of *negative traces* denoted by  $\mathcal{D}_{LTL}^{neg}$  (violating the true LTL formula  $\phi$ ), where  $\mathcal{D}_{LTL}^{pos} \cap \mathcal{D}_{LTL}^{neg} = \emptyset$ . For an arbitrary LTL formula  $\phi'$ , we write that  $\mathcal{D}_{LTL}$  is *consistent* with  $\phi'$  if the traces in  $\mathcal{D}_{LTL}^{pos}$  satisfy  $\phi'$  and the traces in  $\mathcal{D}_{LTL}^{neg}$  violate  $\phi'$ . Algorithm 1 illustrates the framework that we use to learn  $\phi_I$  where the upper bound on the size of the true LTL formula, denoted by  $Itr$ , is given.

In Algorithm 1, we convert the problem of inferring LTL formulas from  $\mathcal{D}_{LTL}$  to a satisfiability problem in the propositional logic. The satisfiability problem is about assessing whether a logical formula is satisfiable or not. For the satisfiability problem in the Boolean domain, we can use *Boolean*

*satisfiability* (SAT) solvers. One of the common SAT solvers is the *Z3 theorem prover* [69].

We use propositional formulas in the satisfiability problem to infer LTL formulas. If  $\mathcal{P}$  is a set of propositional variables in the Boolean domain, then a propositional variable  $\rho \in \mathcal{P}$  is a propositional formula. In addition, if  $\Phi$  and  $\Psi$  are propositional formulas, then  $\neg\Phi$  and  $\Phi \vee \Psi$  are propositional formulas as well. Then, we define a model<sup>1</sup> of a propositional formula as a mapping  $\gamma : \mathcal{P} \rightarrow \mathbb{B}$ . The semantics of this propositional valuation is given by a satisfaction relation that is defined as follows.  $\gamma \models \rho$  if and only if  $\gamma(\rho) = 1$ ,  $\gamma \models \neg\Phi$  if and only if  $\gamma \not\models \Phi$ ,  $\gamma \models \Phi \vee \Psi$  if and only if  $\gamma \models \Phi$  or  $\gamma \models \Psi$ , and finally, if  $\gamma \models \Phi$ , then we introduce  $\gamma$  as a *model* of  $\Phi$ . In the case that such a model  $\gamma$  exists, the propositional formula  $\Phi$  is satisfiable. Such a model provides us with sufficient information to construct an LTL formula from that model. To infer LTL formulas using propositional formulas, we use the framework proposed in [59] to construct a propositional formula of a given size that is consistent with the dataset of traces  $\mathcal{D}_{LTL}$ . By exploiting the fact that for a propositional formula, multiple distinguished models can be found, we can construct a distinct LTL formula from each model. Hence, the key idea in Algorithm 1 is to infer all possible LTL formulas with the maximum size of  $Itr$  from  $\mathcal{D}_{LTL}$  using propositional formulas.

In Algorithm 1,  $\mathcal{P}$  contains three types of propositional variables: 1)  $p_{j,\delta}$  where  $j \in \{1, \dots, k\}$  and  $\delta \in \Sigma \cup \mathcal{C}$ ; 2)  $\ell_{j,j'}$  where  $j \in \{1, \dots, k\}$  and  $j' \in \{1, \dots, j-1\}$ ; and  $r_{j,j'}$  where  $j \in \{1, \dots, k\}$  and  $j' \in \{1, \dots, j-1\}$ . Here,  $k$  denotes the size of the propositional formula  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$ . The variables  $p_{j,\delta}$  encode a labeling of a syntax DAG from which we construct an LTL formula in the sense that if a variable  $p_{j,\delta}$  is set to true, then node  $j$  is labeled with  $\delta$ . The variables  $\ell_{j,j'}$  and  $r_{j,j'}$  encode the structure of the syntax DAG (i.e., the left and/or right child of inner nodes) (cf. [59] for details).

Algorithm 1 starts from  $k = 1$  and constructs a propositional formula  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  consistent with  $\mathcal{D}_{LTL}$  where  $k$  is the size of the propositional formula (Line 5 in Algorithm 1). Then Algorithm 1 computes a model  $\gamma$  of  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$ , if  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  is satisfiable (Line 7 in Algorithm 1).  $\gamma$  maps a subset of propositional variables in  $\mathcal{P}$  to *True* denoted by  $Var_\gamma^1$  and maps the rest of propositional variables to *False* denoted by  $Var_\gamma^0$ . In this way,  $\gamma$  can be used to construct an LTL formula by constructing a DAG corresponding to that LTL formula. In order to compute all the possible models of  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$ , we add the constraint in Line 9 to ensure that an already computed  $\gamma$  is not computed again. In Line 9,  $\bigwedge_{k'=1, \dots, |E_\gamma|} \bigvee_{\rho \in Var_{\gamma_{k'}}^1} (\rho \notin Var_{\gamma_{k'}}^1)$  means that, in the model  $\gamma$  that is going to be computed in the next iteration  $i + 1$ , at least one of the propositional variables that is going to be set to *True* must be different from the ones that were set to true in the previously computed models. In Algorithm 1,  $E_{LTL}$  is the set containing all the inferred LTL formulas with the maximum size of  $Itr$  where we construct  $\phi_I$

<sup>1</sup>The term “model” in Section V refers to a model of a propositional formula and is different from the system model used in the rest of the article.

by forming a disjunction among the formulas in  $E_{LTL}$ .  $E_{\gamma}$  is an auxiliary set containing all the models of  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  to track the models that are already computed and to block/prevent them from being computed again.

If  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  of size  $k$  is not satisfiable, then  $k$  increases by 1 to infer LTL formulas using the propositional formula  $\Phi_{k+1}^{\mathcal{D}_{LTL}, \gamma}$  of size  $k+1$  (Lines 17 to 20 in Algorithm 1). The outer *while* loop in Algorithm 1 terminates once  $N_o$  possible LTL formulas with the maximum size of  $Itr$  are inferred from the given dataset of traces  $\mathcal{D}_{LTL}$ . Finally, we construct  $\phi_I := \bigvee_{i=1, \dots, |E_{LTL}|} \phi^i$ .

**Theorem 1:** Given a traces dataset  $\mathcal{D}_{LTL}$ , if  $E_{LTL} \neq \emptyset$ , then  $\phi \rightarrow \phi_I$  holds for  $\phi_I := \bigvee_{i=1, \dots, |E_{LTL}|} \phi^i$ , where the size of each  $\phi^i \in E_{LTL}$  is at most  $Itr$  and  $\phi$  is the true LTL formula.

*Proof:* Algorithm 1 terminates when all the possible LTL formulas with the maximum size of  $Itr$  are inferred including the true LTL formula  $\phi$ , all consistent with the given dataset  $\mathcal{D}_{LTL}$ . Hence, we conclude that if  $E_{LTL} \neq \emptyset$ , then  $\phi \rightarrow \phi_I$  holds for  $\phi_I := \bigvee_{i=1, \dots, |E_{LTL}|} \phi^i$  where the size of each  $\phi^i \in E_{LTL}$  is at most  $Itr$ . ■

## B. REDUCING THE SIZE OF THE INFERRED LTL FORMULA

In this subsection, we present a novel framework and its associated algorithm for reducing the size of inferred LTL formula  $\phi_I$  such that  $\phi \rightarrow \phi_I$  still holds. The reason is that reducing the size of inferred LTL formula while satisfying  $\phi \rightarrow \phi_I$  drastically decreases the computational cost in model discrimination. In what follows, we first introduce *prior information* to constrain the size of inferred LTL formula such that  $\phi \rightarrow \phi_I$ . Then, we propose an algorithm to further reduce the size of the inferred formula while satisfying  $\phi \rightarrow \phi_I$ .

### 1) PRIOR INFORMATION

Here, we introduce *prior information* that we use for reducing the size of inferred LTL formula  $\phi_I$ , when it is available. Before introducing *prior information*, we need to explain how we use CFG to generate strings of LTL formulas according to a given syntax DAG.

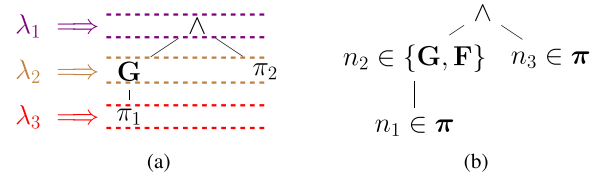
As mentioned earlier, an LTL formula can be represented as a syntax DAG. This DAG can be encoded as a CFG. In defining the productions in CFG corresponding to a DAG, the head of the first production corresponds to the root node and the head of the last production corresponds to the outermost child node in a DAG.

**Example 2:** The DAG corresponding to the LTL formula  $(\mathbf{G} \pi_1) \wedge \pi_2$  is shown in Fig. 3(a) with the corresponding CFG defined in (20), where  $\lambda_1$  corresponds to the root node  $\wedge$  and  $\lambda_3$  corresponds to the outermost child node  $\pi_1$ . Further, each layer in the DAG corresponds to a non-terminal  $\lambda_i \in \mathcal{N} = \{\lambda_1, \lambda_2, \lambda_3\}$  and  $\Pi = \{\mathbf{G}, \pi_1, \pi_2\}$ :

$$\lambda_1 \Rightarrow \lambda_2 \wedge \lambda_2, \lambda_2 \Rightarrow \mathbf{G} \lambda_3 \mid \pi_2, \lambda_3 \Rightarrow \pi_1. \quad (20)$$

Now, we define *prior information* in the following.

**Definition 5:** The prior information  $PI$  represents a DAG structure with the following properties.



**FIGURE 3. (a) The DAG corresponding to the LTL formula  $(\mathbf{G} \pi_1) \wedge \pi_2$  in Example 2. Each layer in the DAG corresponds to a non-terminal  $\lambda_i \in \mathcal{N} = \{\lambda_1, \lambda_2, \lambda_3\}$ . (b) The DAG structure representing a given  $PI, CFG_{PI}$  corresponding to the given  $PI$  is defined in (21) with  $\pi \triangleq \{\pi_1, \pi_2\}$ .**

- 1) The labels of some of the nodes are known and are **fixed**, and the labels of the rest of the nodes are to be inferred which are referred to as **unfixed** nodes.
- 2) The label of each **unfixed** node belongs to a predetermined set of Boolean connectives, temporal operators, and atomic propositions.
- 3) The size of  $PI$  equals  $Itr$  which is the given upper bound on the size of the true LTL formula  $\phi$ .
- 4) The DAG corresponding to the true LTL formula  $\phi$  is **consistent with the DAG structure or a partial DAG structure of  $PI$** . We explain the concept of being consistent with the DAG structure or the partial DAG structure of  $PI$  through Example 3. For simplicity, we write an LTL formula  $\phi$  is *consistent* with  $PI$  as an equivalent form of stating that the DAG corresponding to the LTL formula  $\phi$  is consistent with the DAG structure or a partial DAG structure of  $PI$ .

Determining the DAG structure of  $PI$ , the fixed nodes, and the predetermined sets associated with unfixed nodes depends on the specific scenario at hand. Hence,  $PI$  should be determined according to a certain scenario by a domain expert. For implementing  $PI$ , we use CFG which allows us to encode  $PI$  such that we can also infer LTL formulas that are *consistent* with the partial structure of  $PI$ . This is specifically beneficial since we do not know the size of the true LTL formula. We use  $CFG_{PI}$  to denote the CFG corresponding to a given  $PI$ . Here, for the set of non-terminals, we have  $\mathcal{N} = \{\lambda_1, \lambda_2, \dots, \lambda_{N_k}\}$ . For the set of terminals, we have  $\Pi \subseteq \Sigma \cup \mathcal{O}$ .

**Example 3:** Fig 3(b) illustrates the DAG structure representing a given  $PI$ . We define the CFG corresponding to the given  $PI$  with the productions defined in (21). Here  $\Pi = \{\mathbf{G}, \mathbf{F}, \wedge, \pi_1, \pi_2, \pi_3\}$  and  $\mathcal{N} = \{\lambda_1, \lambda_2, \lambda_3\}$ .

$$\begin{aligned} \lambda_1 &\Rightarrow \lambda_2 \wedge \lambda_2 \\ \lambda_2 &\Rightarrow \mathbf{G} \lambda_3 \mid \mathbf{F} \lambda_3 \mid \pi_1 \mid \pi_2 \mid \pi_3 \\ \lambda_3 &\Rightarrow \pi_1 \mid \pi_2 \mid \pi_3. \end{aligned} \quad (21)$$

Possible LTL formulas that can be generated from (21) are  $(\mathbf{F} \pi_1) \wedge \pi_2$ ,  $(\mathbf{G} \pi_1) \wedge \pi_2$ ,  $\pi_1 \wedge \pi_2$ ,  $\mathbf{G} \pi_1$ ,  $\mathbf{F} \pi_3$ , etc. Note that using CFG for implementing  $PI$  may result in generating LTL formulas that are not consistent with the given  $PI$ . For example, the LTL formula  $\pi_1 \wedge \pi_2$  is not consistent with  $PI$ , since the labels of neither of the children node of  $\wedge$  belong to  $\{\mathbf{G}, \mathbf{F}\}$ . The LTL formula  $\mathbf{G} \pi_1$  is consistent with partial

structure of  $\mathcal{PI}$ . The LTL formula  $(\mathbf{F} \pi_1) \wedge \pi_2$  is consistent with the whole DAG structure of  $\mathcal{PI}$ .

*Remark 2:* The set of LTL formulas consistent with  $\mathcal{PI}$  is a subset of the LTL formulas that can be generated by  $CFG_{\mathcal{PI}}$ .

## 2) CONSTRAINING THE INFERENCE OF LTL FORMULAS USING PRIOR INFORMATION

Here, we explain how we use prior information  $\mathcal{PI}$  for reducing the size of  $\phi_I$ . In the first step, in Algorithm 1, we add  $CFG_{\mathcal{PI}}$  as a set of constraints to  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  in Line 5 in Algorithm 1. In this way, if  $\Phi_k^{\mathcal{D}_{LTL}, \gamma}$  is satisfiable in Line 6 in Algorithm 1, then each computed model  $\gamma$  in Line 7 in Algorithm 1 satisfies  $CFG_{\mathcal{PI}}$ . Consequently, each constructed LTL formula  $\phi^i$  in Line 9 in Algorithm 1 is consistent with  $CFG_{\mathcal{PI}}$ . If any LTL formulas consistent with  $CFG_{\mathcal{PI}}$  are inferred, then we store them in the set  $E_{CFG}$ . Note that  $E_{CFG} \subseteq E_{LTL}$ .

As explained in Remark 2, it is possible that not all the LTL formulas in  $E_{CFG}$  are consistent with  $\mathcal{PI}$ . Hence, in the second step, if  $E_{CFG} \neq \emptyset$ , then we keep only those formulas that are consistent with  $\mathcal{PI}$  and store them in  $E_{\mathcal{PI}}$ . Finally, we form  $\phi_I := \bigvee_{i=1, \dots, |E_{\mathcal{PI}}|} \phi^i$  where  $\phi^i \in E_{\mathcal{PI}}$ .

*Proposition 4:* Given dataset of traces  $\mathcal{D}_{LTL}$  and  $\mathcal{PI}$ , if  $E_{CFG} \neq \emptyset$ , then  $\phi \rightarrow \phi_I$  holds for  $\phi_I := \bigvee_{i=1, \dots, |E_{\mathcal{PI}}|} \phi^i$  where the size of each  $\phi^i \in E_{\mathcal{PI}}$  is at most  $Itr$  and  $\phi$  is the true LTL formula.

*Proof:* 1) We know that Algorithm 1 terminates when all the possible LTL formulas with the maximum size of  $Itr$  are inferred, all consistent with the given dataset  $\mathcal{D}_{LTL}$  and  $CFG_{\mathcal{PI}}$ . For  $E_{\mathcal{PI}}$ , we also have  $E_{\mathcal{PI}} \subseteq E_{CFG}$  which implies that the size of all the LTL formulas in  $E_{\mathcal{PI}}$  is at most  $Itr$  (See Remark 2). 2) If  $E_{CFG} \neq \emptyset$ , we obtain  $E_{\mathcal{PI}}$  which contains all the LTL formulas that satisfy the third property of  $\mathcal{PI}$ . Note that it is assumed that the third property mentioned in Definition 5 is already satisfied in the given  $\mathcal{PI}$ .

Using 1) and 2), we conclude that if  $E_{CFG} \neq \emptyset$ , then  $\phi \rightarrow \phi_I$  holds for  $\phi_I := \bigvee_{i=1, \dots, |E_{\mathcal{PI}}|} \phi^i$  where the size of each  $\phi^i \in E_{\mathcal{PI}}$  is at most  $Itr$ . ■

## 3) REDUCING THE NUMBER OF THE INFERRED LTL FORMULAS

For further increasing computational efficiency (e.g., in model discrimination), we can reduce the size of the inferred formulas in both  $E_{LTL}$  and  $E_{\mathcal{PI}}$  using Algorithm 2. In Algorithm 2, for each two distinct  $\phi^i$  and  $\phi^{i'} \in E_{Inf}$ ,  $E_{Inf} \in \{E_{LTL}, E_{\mathcal{PI}}\}$ , we check 1) if  $\phi^i \rightarrow \phi^{i'}$  and 2) if  $\phi^{i'} \rightarrow \phi^i$  (Lines 5 and 6 in Algorithm 2). For verifying whether  $\phi^i \rightarrow \phi^{i'}$ , we first convert both formulas to deterministic finite automata (DFA)  $\mathcal{A}$  and  $\mathcal{A}'$ , respectively. Then, we check whether the language of  $\mathcal{A}$  is included in the language of  $\mathcal{A}'$ . We kindly refer the reader to [40] for a detailed description of how to verify the implication between two LTL formulas. If 1) holds and 2) does not hold, we keep  $\phi^i$  and add it to  $E'_{Inf}$ . If both 1) and 2) hold and  $\phi^{i'}$  is already stored in  $E'_{Inf}$ , we ignore  $\phi^{i'}$  (Lines 6 to 10 in

### Algorithm 2: Size Reduction of the Inferred Formula.

---

**Input:** Inferred formula set  $E_{Inf} \in \{E_{LTL}, E_{\mathcal{PI}}\}$

```

1  $E'_{Inf} = \emptyset$ 
2 if  $E_{Inf} \neq \emptyset$  then
3   for  $i = 1, \dots, |E_{Inf}|$  do
4     for  $i' = 1, \dots, |E_{Inf}|$  do
5       if  $i \neq i'$  and  $\phi^i \rightarrow \phi^{i'}$  then
6         if  $\phi^{i'} \rightarrow \phi^i$  and  $\phi^{i'} \in E'_{Inf}$  then
7           Continue
8         else
9           Add  $\phi^i$  to  $E'_{Inf}$ 
10          Break
11        end if
12      end if
13    end for
14  end for
15  $E_{Red} = E_{Inf} / E'_{Inf}$ 
16 if  $E_{Red} \neq \emptyset$  then
17    $\psi_I \leftarrow \bigvee_{i=1, \dots, |E_{Red}|} \phi^i$ 
18 else
19    $\psi_I \leftarrow \phi_I$ 
20 end if
21 return  $\psi_I$ 

```

---

Algorithm 2). Then, we form the set  $E_{Red} = E_{Inf} / E'_{Inf}$  using which we obtain  $\psi_I := \bigvee_{i=1, \dots, |E_{Red}|} \phi^i$  where  $\phi^i \in E_{Red}$ .

*Proposition 5:* In Algorithm 2, if  $E_{Red} \neq \emptyset$ , then we obtain  $\phi \rightarrow \psi_I := \bigvee_{i=1, \dots, |E_{Red}|} \phi^i$  where  $\phi^i \in E_{Red}$  and  $\phi$  is the true LTL formula, and the size of  $\psi_I$  is smaller than or equal to the size of  $\phi_I$ .

*Proof:* In Algorithm 2,  $E_{Red}$  contains all the LTL formulas  $\phi^i \in E_{Inf}$ , such that  $\phi^{i'} \rightarrow \phi^i$  holds for at least one  $\phi^{i'} \in E_{Inf}$  with  $i \neq i'$ . This implies that  $\phi_I \rightarrow \psi_I$  where  $\phi_I := \bigvee_{i=1, \dots, |E_{Inf}|} \phi^i$  with  $\phi^i \in E_{Inf}$  and  $\psi_I := \bigvee_{k=1, \dots, |E_{Red}|} \phi^k$  with  $\phi^k \in E_{Red}$ . We also can conclude that  $E_{Red} \subseteq E_{Inf}$  and hence  $|E_{Red}| \leq |E_{Inf}|$  from which follows that the size of  $\psi_I$  is smaller than or equal to the size of  $\phi_I$ . Moreover, since we already know that  $\phi \rightarrow \phi_I$  and  $\phi_I \rightarrow \psi_I$ , we can write  $\phi \rightarrow \psi_I$ . ■

## VI. 7-DISTINGUISHABILITY AND MODEL INVALIDATION

After obtaining the inferred system models and LTL formulas, we then propose a detectability analysis algorithm for  $T$ -distinguishability (i.e., to solve Problem 2), and the (guaranteed) detection time  $T$  is found by solving the problem below with increasing  $T$ :

*Theorem 2 (T-Distinguishability):* A pair of learned piecewise constrained affine inclusion models  $\mathcal{H}^i$  and  $\mathcal{H}^j$ ,  $i \neq j$ , with LTL formulas  $\phi_I^i$  and  $\phi_I^j$  is  $T$ -distinguishable if the following is infeasible for any  $t_0$ :

$$\text{Find } x_t^*, v_t^*, u_t, y_t, s_{t,*}^*, \tilde{s}_{t,\dagger}^*, a_{t,*}^*, \tilde{a}_{t,\dagger}^*, c_t^{\sigma^m}, b_t^{\sigma^m}$$

$$\text{s.t. } \forall \star \in \{i, j\}, \sigma^m \in \{e_1, \dots, e_{|\Sigma_m|}\}, * \in \mathbb{Z}_1^{q_{\sigma^m}^{f,*}},$$

$$\ell \in \mathbb{Z}_1^{|\Sigma_s|}, \sigma^{s,\ell} \in \{0, 1\}, \dagger \in \mathbb{Z}_1^{q_{\sigma^{s,\ell}}^{g,*}}, t \in \mathbb{Z}_1^{t_0+T-1}:$$

$$x_{t+1}^* \leq \bar{A}_{\sigma^m,*}^* x_t^* + \bar{B}_{\sigma^m,*}^* u_t + \bar{h}_{\sigma^m,*}^{f,*} + (s_{t,*}^* + c_t^{\sigma^m}) \mathbb{1}, \quad (22a)$$

$$x_{t+1}^* \geq \underline{A}_{\sigma^m,*}^* x_t^* + \underline{B}_{\sigma^m,*}^* u_t + \underline{h}_{\sigma^m,*}^{f,*} - (s_{t,*}^* + c_t^{\sigma^m}) \mathbb{1}, \quad (22b)$$

$$S_{\sigma^m,*}^{x,*} x_t^* + S_{\sigma^m,*}^{u,*} u_t \leq \beta_{\sigma^m,*}^* + (s_{t,*}^* + c_t^{\sigma^m}) \mathbb{1}, \quad (22c)$$

$$y_t = Cx_t^* + v_t^*, \quad (22d)$$

$$x_{t,\ell}^{c,*} \leq \bar{H}_{\sigma^{s,\ell},\dagger}^{s,*} z_{t,\ell}^* + \bar{h}_{\sigma^{s,\ell},\dagger}^{g,*} + (s_{t,\dagger}^* + c_t^{\sigma^{s,\ell}}) \mathbb{1}, \quad (22e)$$

$$x_{t,\ell}^{c,*} \geq \underline{H}_{\sigma^{s,\ell},\dagger}^{s,*} z_{t,\ell}^* + \underline{h}_{\sigma^{s,\ell},\dagger}^{g,*} - (s_{t,\dagger}^* + c_t^{\sigma^{s,\ell}}) \mathbb{1}, \quad (22f)$$

$$S_{\sigma^{s,\ell},\dagger}^{z,*} z_{t,\ell}^* \leq \beta_{\sigma^{s,\ell},\dagger}^* + (s_{t,\dagger}^* + c_t^{\sigma^{s,\ell}}) \mathbb{1}, \quad (22g)$$

$$a_{t,*}^* \in \{0, 1\}, \tilde{a}_{t,\dagger}^* \in \{0, 1\}, b_t^{\sigma^m} \in \{0, 1\}, b_t^{\sigma^{s,j}} \in \{0, 1\}, \quad (22h)$$

$$\begin{aligned} & \text{SOS-1: } (a_{t,*}^*, s_{t,*}^*), \text{ SOS-1: } (\tilde{a}_{t,\dagger}^*, \tilde{s}_{t,\dagger}^*), \\ & \text{SOS-1: } (b_t^{\sigma^m}, c_t^{\sigma^m}), \text{ SOS-1: } (b_t^{\sigma^{s,j}}, c_t^{\sigma^{s,j}}), \end{aligned} \quad (22i)$$

$$\sum_{\xi=1}^{q_{\sigma^m}^{f,*}} a_{t,\xi}^* = 1, \sum_{\xi=1}^{q_{\sigma^m}^{g,s,j}} \tilde{a}_{t,\xi}^* = 1, \sum_{\sigma^m \in e_1} b_t^{\sigma^m} = 1, \quad (22j)$$

$$w_t^* \in \mathcal{W}, v_t^* \in \mathcal{V}, u_t \in \mathcal{U}, x_t^* \in \mathcal{X}, \quad (22k)$$

$$\{\sigma_t^m, \sigma_t^{s,j}\}_{t=\underline{t}}^{\bar{t}} \in V_T(\phi_I^*), \quad (22l)$$

where SOS-1 refers to Special Ordered Set of type 1 (i.e., at most one member of the set can be non-zero [70]),  $s_{t,*}^*, \tilde{s}_{t,\dagger}^*$  are slack variables and  $c_t^{\sigma^m}$  is vector of slack variables for SOS-1 constraints,  $c_t^{\sigma^m} = 1$  corresponds to  $\sigma_t^m$  being true and 0 otherwise,  $\underline{t} = t_0 - T_m + 1$  and  $\bar{t} = t_0 + T + T_m - 1$  determines the horizon for  $\sigma_t^m, \sigma_t^{s,j}$ , where  $T_m$  is the maximum length of sampled traces (data)<sup>2</sup>, and the set of valid subtraces  $V_T(\phi_I^*)$  is constructed recursively using encoding of the LTL formula  $\phi_I^*$  according to Section II (cf. (4)–(10)).

*Proof:*  $a_{t,*}^* = 1$  and  $b_t^{\sigma^m} = 1$  imply that (22a)–(22c) hold, since the SOS-1 constraints in (22i) ensure that  $s_{t,*}^* = 0$  and  $c_t^{\sigma^m} = 0$  correspondingly. On the contrary, if  $a_{t,*}^* = 0$  and/or  $b_t^{\sigma^m} = 0$ , it means that  $s_{t,*}^*$  and/or  $c_t^{\sigma^m}$  are free and then (22a)–(22c) hold trivially. Similarly,  $\tilde{a}_{t,\dagger}^* = 1$  and  $b_t^{\sigma^{s,j}} = 1$ , or  $\tilde{a}_{t,\dagger}^* = 0$  and/or  $b_t^{\sigma^{s,j}} = 0$  imply that (22e)–(22g) hold. In addition, (22j) ensures that, at each time step  $t$ , only one partition is valid for each of the state and output equations, and only one switching signal/mode (for  $\sigma^m \in \Sigma_m$ ) is valid. Thus, if the above problem is infeasible, it means that there exists no common behavior that is satisfied by both models, i.e.,  $\hat{\mathcal{S}}^i \cap \hat{\mathcal{S}}^j = \emptyset$ ; hence, the pair of models is distinguishable from each other. ■

The detection time  $T_0$  for all models can then be selected as the maximum value of  $T$  for every pair of the given model.

<sup>2</sup> Although LTL formulas are usually unbounded, we can use the length of the data used to infer the LTL formula (in Section V) to determine  $\underline{t}$  and  $\bar{t}$  that bidirectionally extend the time horizon based on the “LTL bound.”

Next, we introduce a data-driven model invalidation algorithm (cf. Algorithm 3) that allows us to rule out/eliminate all models that are incompatible with newly observed data, i.e., to address Problem 3. If not all model pairs are  $T$ -distinguishable (according to Theorem 2), Algorithm 3 will still return a set of all models that are consistent with the input-mode-output data up to the current time step  $t_c$ . Specifically, we will solve Problem 3 to check if the newly observed data are consistent with the learned/inferred model-task pairs: In Algorithm 3, we initially verify in Line 4 if the newly observed mode sequence/trace satisfies the inferred formula  $\phi_I^l$ , and if it does, we proceed to check the compatibility of the learned models  $\mathcal{H}^l$  with newly observed trajectory data using the following result in Line 9:

*Theorem 3:* Given a learned constrained piecewise affine inclusion model  $\mathcal{H}^l$  and a (newly observed) length- $T$  input-mode-output sequence  $\{u_t, \sigma_t, y_t\}_{t=t_c-T+1}^{t_c}$  at time  $t_c$ , the model is invalidated if the following problem is infeasible for any  $t_c \in \mathbb{Z}_0^\infty$ :

Find  $x_t, v_t, s_{t,*}, a_{t,*}, s_{t,\dagger}, \tilde{a}_{t,\dagger}$

s.t.  $\forall t \in \mathbb{Z}_{t_c}^{t_c+T-1}, * \in \mathbb{Z}_1^{q_{\sigma_t^m}^{f,l}}, k \in \{\ell \mid \sigma_t^{s,\ell} = 1\}, \dagger \in \mathbb{Z}_1^{q_{\sigma_t^{s,k}}^{g,l}} :$

$$x_{t+1} \leq \bar{A}_{\sigma_t^m,*}^l x_t + \bar{B}_{\sigma_t^m,*}^l u_t + \bar{h}_{\sigma_t^m,*}^{f,l} + s_{t,*} \mathbb{1}, \quad (23a)$$

$$x_{t+1} \geq \underline{A}_{\sigma_t^m,*}^l x_t + \underline{B}_{\sigma_t^m,*}^l u_t + \underline{h}_{\sigma_t^m,*}^{f,l} - s_{t,*} \mathbb{1}, \quad (23b)$$

$$S_{\sigma_t^m,*}^{x,l} x_t + S_{\sigma_t^m,*}^{u,l} u_t \leq \beta_{\sigma_t^m,*}^l + s_{t,*} \mathbb{1}, \quad (23c)$$

$$y_t = Cx_t + v_t, \quad v_t \in \mathcal{V}, \quad x_t \in \mathcal{X}, \quad (23d)$$

$$x_{t,k}^c \leq \bar{H}_{\sigma_t^{s,k},\dagger}^{s,k} z_{t,k} + \bar{h}_{\sigma_t^{s,k},\dagger}^{g,l} + \tilde{s}_{t,\dagger} \mathbb{1}, \quad (23e)$$

$$x_{t,k}^c \geq \underline{H}_{\sigma_t^{s,k},\dagger}^{s,k} z_{t,k} + \underline{h}_{\sigma_t^{s,k},\dagger}^{g,l} - \tilde{s}_{t,\dagger} \mathbb{1}, \quad (23f)$$

$$S_{\sigma_t^{s,k},\dagger}^{z,l} z_{t,k} \leq \beta_{\sigma_t^{s,k},\dagger}^l + \tilde{s}_{t,\dagger} \mathbb{1}, \quad (23g)$$

$$a_{t,*} \in \{0, 1\}, \text{ SOS-1: } (a_{t,*}, s_{t,*}), \sum_{\xi=1}^{q_{\sigma_t^m}^{f,l}} a_{t,\xi} = 1, \quad (23h)$$

$$\tilde{a}_{t,\dagger} \in \{0, 1\}, \text{ SOS-1: } (\tilde{a}_{t,\dagger}, \tilde{s}_{t,\dagger}), \sum_{\xi=1}^{q_{\sigma_t^{s,k}}^{g,l}} \tilde{a}_{t,\xi} = 1. \quad (23i)$$

*Proof:* The construction follows similar steps to Theorem 2, but with only one model and a given/measured mode sequence. ■

If either of the checks is infeasible, then that learned model-task pair can be ruled out as incompatible.

However, when the size of the inferred formulas is large (even after applying the reduction approach using Algorithm 2 in Section V-B3), the resulting optimization problems in both Theorems 2 and 3 may also be large and this results in high computational cost and large memory requirements. To remedy this, in practice, since the  $\vee$  condition can be considered

**Algorithm 3:** Model Discrimination with Length  $T$ .

---

**Input:** Models  $\{\mathcal{G}^l\}_{l=1}^{N_m}$ ,  
Input-Output Sequence  $= \{u_t, \sigma_t, y_t\}_{t=t_0-T+1}^{t_0}$

```

1 function findModel( $\{\mathcal{G}^l\}_{l=1}^{N_m}, \{u_t, \sigma_t, y_t\}_{t=t_0-T+1}^{t_0}$ )
2   valid  $\leftarrow \{\mathcal{G}^l\}_{l=1}^{N_m}$ ;
3   for  $l = 1 : N_m$  do
4     Check  $\sigma \models \phi_l^l$ ;
5     if infeasible then
6       Remove  $l$  from valid;
7       Continue;
8     end if
9     Check Feasibility of Theorem 3;
10    if infeasible then
11      Remove  $l$  from valid;
12    end if
13  end for
14  return valid

```

---

separately, we can instead consider all possible combinations of the subformulas in parallel and the maximum  $T$  among all these combinations will be considered as the solution to Problem 2, while for Problem 3, the problem in Theorem 3 is infeasible if the subproblems with all combinations of subformulas are all infeasible. For instance, for a model-task pair with inferred formulas  $\phi_l^i = \bigvee_{k=1}^{k=|E_i|} \phi_k^i$  and  $\phi_l^j = \bigvee_{k=1}^{k=|E_j|} \phi_k^j$ , we can consider all combinations of subformulas of these pairs of formulas, i.e., the subformula pairs  $(\phi_k^i, \phi_k^j)$ ,  $\forall k = \mathbb{Z}_1^{|E_i|}, k' \in \mathbb{Z}_1^{|E_j|}$ .

**VII. SIMULATIONS**

The simulations in Sections VII-A and VII-B are implemented in MATLAB 2021a with the Gurobi optimization solver [70] on a 2.2 GHz machine with 16 GB RAM.

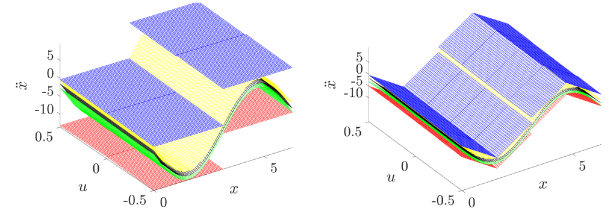
**A. SINGLE LINK ROBOT ARM LINK**

First, we tested our algorithms using a single link robot arm example with the following dynamics [71]:

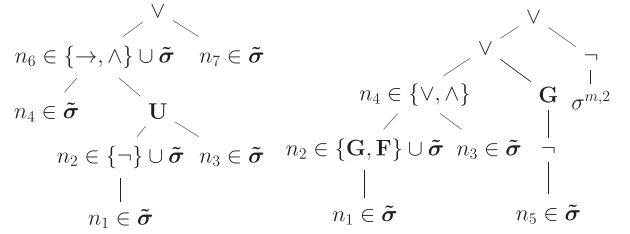
$$\ddot{x}(t) = -\frac{m^k g l^k}{J^k} \sin(x(t)) - \frac{D^k}{J^k} \dot{x}(t) + \frac{1}{J^k} u(t), \quad (24)$$

where  $x(t)$  is the angle of the arm,  $u(t)$  is the control input, and for each discrete state/mode  $q^k = (m^k, l^k, J^k, D^k)$ ,  $m^k$  is the mass,  $l^k$  is the length of the arm that can extend and retract (making the robot arm a switched system),  $J^k$  is the moment of inertia, and  $D^k$  is the damping coefficient. These parameters  $q^k$  change depending on the angle parameter  $x(t)$ . We discretized the system with sampling time  $\delta t = 0.1$  s and select the  $u(t) = 0.5 \cos(t)$ .

For the first model, the ground truth formula for the task specification is  $\phi^1 = (\sigma^{m,1} \rightarrow (\neg \sigma^{m,2} \mathbf{U} \sigma^{m,1})) \vee \sigma^{m,3}$  where  $\sigma^{m,k}$  is an atomic proposition for  $\sigma_t^m = e_k$  and  $q^k$  can be one of three values:  $q^1 = (1, 1, 2, 2)$ ,  $q^2 = (1, 1.5, 3, 475, 2)$ , or  $q^3 = (1, 2, 8, 2)$ . The second model is with the ground truth formula  $\phi^2 = (\mathbf{G} \sigma^{m,1} \vee \sigma^{m,2}) \vee \mathbf{G} \neg \sigma^{m,3}$  and  $q^k$  can be one of three values:  $q^1 = (2, 1, 2, 2)$ ,  $q^2 = (2, 1.5, 3, 475, 2)$ , or  $q^3 = (2, 2, 8, 2)$ .



**FIGURE 4.** Illustration of the learned over-approximation model with different number of partitions for the function in (24) with  $q^k = (1, 1, 2, 2)$  in example A.



**FIGURE 5.** The DAG structure representing the given  $PI$  for the model-task pair  $(\mathcal{G}^1, \phi^1)$  (left) and  $(\mathcal{G}^2, \phi^2)$  (right) in example A with  $\tilde{\sigma} \triangleq \{\sigma^{m,1}, \sigma^{m,2}, \sigma^{m,3}\}$ .

**TABLE 1.** Comparison of the guaranteed detection times  $T$  and CPU times (s) for model discrimination (cf. Theorem 2) with and without inferred LTL formulas in example A.

	$T$	CPU Time (s)
(i) Without formulas	17	165.02
(ii) With formulas	17	403.20

We first generated 200 random data points for each mode  $\begin{bmatrix} \hat{x}_{t+1} \\ \hat{x}_t \end{bmatrix} = f^l(\begin{bmatrix} \hat{x}_t \\ u_t \end{bmatrix})$  of both time-discretized models and used this data to learn the inclusion/over-approximated dynamics model as shown in Fig. 4, where the true model is included/contained within the hyperplanes of learned dynamics models, as expected, and the more partitions we consider, the better the model accuracy will be.

For each model, we generated 50 positive traces and 50 negative traces with three modes, all with a maximum length of 20 time steps. To infer  $\phi_l^1$ , we used the prior information  $PI$  in Fig. 5 (left) and obtained  $\phi_l^1 = (\sigma^{m,1} \vee ((\sigma^{m,1} \mathbf{U} \sigma^{m,1}) \vee \sigma^{m,3}) \vee (\sigma^{m,1} \rightarrow (\sigma^{m,1} \mathbf{U} \sigma^{m,1})) \vee (((\neg \sigma^{m,2} \mathbf{U} \sigma^{m,1}) \wedge \sigma^{m,1}) \vee \sigma^{m,3}) \vee ((\sigma^{m,1} \rightarrow (\neg \sigma^{m,2} \mathbf{U} \sigma^{m,1})) \vee \sigma^{m,3}))$ . To infer  $\phi_l^2$ , we used the prior information  $PI$  depicted in Fig. 5 (right) and obtained  $\phi_l^2 = (\sigma^{m,1} \vee \sigma^{m,2}) \vee (((\mathbf{G} \sigma^{m,1}) \vee \sigma^{m,2}) \vee (\mathbf{G}(\neg \sigma^{m,3}))) \vee (((\sigma^{m,1} \vee \sigma^{m,2}) \vee \mathbf{G}(\neg \sigma^{m,3})) \vee (\neg \sigma^{m,3})) \vee ((\sigma^{m,1} \vee \sigma^{m,2}) \vee \mathbf{G}(\neg \sigma^{m,3}))$ . We reduced the size of the inferred specifications using Algorithm 2, and obtained  $\psi_l^1 = \sigma^{m,1} \rightarrow (\sigma^{m,1} \mathbf{U} \sigma^{m,1})$  and  $\psi_l^2 = ((\sigma^{m,1} \vee \sigma^{m,2}) \vee \mathbf{G}(\neg \sigma^{m,3})) \vee (\neg \sigma^{m,3})$ .

From Table 1, we can see that the guaranteed detection time are the same for all cases, and ‘with formulas’ case requires more computational time when compared to the

**TABLE 2.** Comparison of model discrimination results (with Algorithm 3) for 20 sequences with length  $T = 17$  in example A.

	Mean CPU Time (s)	Discrimination Rate
(i) Dynamics with formulas	2.95	100%
(ii) Only dynamics	3.13	100%
(iii) Only formulas (reduced)	0.02	20%
(iv) Only formulas (all)	0.42	20%

‘without formulas’ case due to the extra integer constraints in the optimization problem when considering the inferred LTL formulas. However, since finding the detection time  $T$  is an off-line process, the CPU time matters less than the CPU time for the model discrimination, which is an online process. As shown in Table 2, the mean CPU time was lower when using both learned/inferred system dynamics and LTL formulas than only using the learned dynamics, while using only the inferred LTL formulas results in only a discrimination rate of 20%.

### B. DUBINS CAR

In this example, we consider a lane change scenario with Dubins Car model [72] as the dynamics model of the vehicle:

$$p_{x,t+1} = p_{x,t} + u_s \cos(\theta_t) \delta t + w_{px,t}, \quad (25a)$$

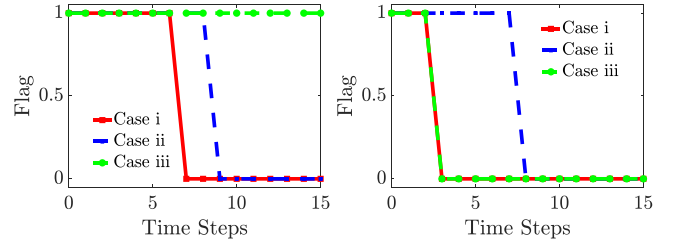
$$p_{y,t+1} = p_{y,t} + u_s \sin(\theta_t) \delta t + w_{py,t}, \quad (25b)$$

$$\theta_{t+1} = \theta_t + \frac{u_s}{L} \tan(u_\phi) \delta t + w_{\theta,t}, \quad (25c)$$

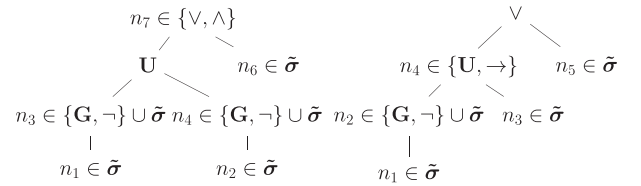
where the system states are  $p_x$  and  $p_y$  that represent the  $(x, y)$ -position of the agent and  $\theta$  as the heading angle of the agent.  $L$  is the length between the front and rear tires and is set to 1.5 m,  $u_s$  is the speed of the agent and is assumed to be 1 m/s, and sampling time  $\delta t$  is set to 0.1 s. The vehicle utilizes for feedback control according to the proportional control law:  $u_\phi = K^k(\theta_{desired} - \theta)$ .

For the first model, the ground truth formula for the system is  $\phi^1 = (\sigma^{m,1} \mathbf{U} \sigma^{m,2}) \wedge \sigma^{s,1}$  and  $K^k$  can take two values:  $K^1 = 0.5$  and  $K^2 = 0.1$  for  $\sigma^{m,1}$  and  $\sigma^{m,2}$ , respectively, with  $\theta_{desired} = 60^\circ$ , while  $\sigma^{m,k}$  is an atomic proposition for  $\sigma_t^m = e_k$  and  $\sigma^{s,1}$  is an atomic proposition for the state constraint  $p_y = \frac{e^{(p_x+6)}}{1+e^{(p_x+6)}}$ . For the second model, the ground truth formula for the system is  $\phi^2 = (\sigma^{m,1} \mathbf{U} \sigma^{m,2}) \vee \sigma^{s,1}$  and  $K^k$  can take two values:  $K^1 = -0.5$  and  $K^2 = -0.1$  for  $\sigma^{m,1}$  and  $\sigma^{m,2}$ , respectively, with  $\theta_{desired} = -60^\circ$ , while  $\sigma^{m,k}$  is an atomic proposition for  $\sigma_t^m = e_k$  and  $\sigma^{s,1}$  is an atomic proposition for the state constraint  $p_y = -\frac{e^{(p_x+6)}}{1+e^{(p_x+6)}}$ .

We first generated 200 random data points for each mode of both models to learn the inclusion model of the system dynamics (not depicted for brevity). Further, we generated 50 positive traces and 50 negative with three modes, all with a maximum length of 20 time steps. Using the prior information  $\mathcal{PI}$  in Fig. 7 (left), we obtained  $\phi^1 = (\sigma^{s,1} \vee (((\mathbf{G} \sigma^{s,1}) \mathbf{U} (\mathbf{G} \sigma^{s,1})) \vee \sigma^{s,1}) \vee ((\sigma^{m,1} \mathbf{U} \sigma^{m,2}) \wedge \sigma^{s,1}) \vee ((\sigma^{m,2} \mathbf{U} (\mathbf{G} \sigma^{s,1})) \vee \sigma^{s,1}))$ . For the second model, using the prior



**FIGURE 6.** Model discrimination results (with Algorithm 3) using two newly observed sequences to invalidate a non-ground truth/false model in example B under three different cases,  $j \in \{i, ii, iii\}$  where Case i leverages both the learned/inferred system dynamics and formulas, Case ii only leverages the learned system dynamics and Case iii only leverages the inferred formulas. Flag  $j$  is 1 when the false model is not (yet) invalidated and is 0 if invalidated.



**FIGURE 7.** The DAG structure representing the given  $\mathcal{PI}$  for the model-task pair  $(\mathcal{G}^1, \phi^1)$  (left) and  $(\mathcal{G}^2, \phi^2)$  (right) in Example B with  $\tilde{\sigma} \triangleq \{\sigma^{m,1}, \sigma^{m,2}, \sigma^{s,1}\}$ .

**TABLE 3.** Comparison of the guaranteed detection times  $T$  and CPU times (s) for model discrimination (cf. Theorem 2) with and without inferred LTL formulas in example B.

	$T$	CPU Time (s)
(i) Without formula	15	132.73
(ii) With formula	15	373.10

**TABLE 4.** Comparison of model discrimination results (with Algorithm 3) for 20 sequences with length  $T = 15$  in example B.

	Mean CPU Time (s)	Discrimination Rate
(i) Dynamics with formulas	2.63	100%
(ii) Only dynamics	3.02	100%
(iii) Only formulas (reduced)	0.07	50%
(iv) Only formulas (all)	1.02	50%

information in Fig. 7 (right), we obtained  $\phi^2 = (\sigma^{m,1} \vee (\sigma^{m,2} \mathbf{U} \sigma^{m,2})) \vee ((\sigma^{m,1} \mathbf{U} \sigma^{m,2}) \vee \sigma^{s,1}) \vee (((\mathbf{G} \sigma^{s,1}) \mathbf{U} \sigma^{m,2}) \vee \sigma^{m,1}) \vee (\sigma^{m,1} \mathbf{U} \sigma^{m,2})$ . We reduced the size of the inferred formulas using Algorithm 2, and obtained  $\psi_I^1 = (\sigma^{m,2} \mathbf{U} (\mathbf{G} \sigma^{s,1})) \vee \sigma^{s,1}$  and  $\psi_I^2 = ((\sigma^{m,1} \mathbf{U} \sigma^{m,2}) \vee \sigma^{s,1}) \vee (((\mathbf{G} \sigma^{s,1}) \mathbf{U} \sigma^{m,2}) \vee \sigma^{m,1})$ .

Similar to the first example, we can see that the guaranteed detection time are the same for all cases from Table 3, while in Table 4, the mean CPU time can be observed to be lower when using both the learned/inferred system dynamics and LTL formulas than when only using the learned dynamics and

the discrimination rate is only 50% if only the inferred formulas are used. We also depicted model discrimination results (with Algorithm 3) in Fig. 6, where two sampled sequences from the ground truth model are used to invalidate learned model-task pairs from a different non-ground truth model. In the figure, we can observe that when using both the inferred system dynamics and LTL formulas (Case i), the non-ground truth model is always invalidated earlier (Flag is 0) than either using only the learned dynamics (Case ii) or only the inferred formulas (Case iii). Moreover, in the left instance, when using only the inferred formulas, the non-ground truth model was not able to be invalidated.

## VIII. CONCLUSION

To address the challenge of data-driven model discrimination for unknown switched systems with unknown LTL specifications, this article first proposed a data-driven set-membership method that over-approximates the unknown dynamics. Then, the article introduced a specification inference method that guarantees that the ground truth LTL is among/included in the inferred formulas, along with a method for reducing the size of the inferred formula when prior information is available. Next, we introduced an optimization-based algorithm to analyze model detectability from noisy, finite data, and a model discrimination algorithm that can rule out learned/inferred model-task pairs that are inconsistent with new observations at run time. The effectiveness of the proposed methods was demonstrated through several illustrative examples. The results indicated that the joint learning/inferring and use of both system dynamics and temporal logic specifications can significantly accelerate the data-driven model discrimination process. Future work will extend the proposed approach to consider inequality constraints in the system model and missing/dropped data.

## REFERENCES

- [1] M. Shvo, A. C. Li, R. T. Icarte, and S. A. McIlraith, "Interpretable sequence classification via discrete optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 9647–9656.
- [2] A. Pnueli, "The temporal logic of programs," in *Proc. 18th Annu. Symp. Foundations Comput. Sci.*, 1977, pp. 46–57, doi: [10.1109/SFCS.1977.32](https://doi.org/10.1109/SFCS.1977.32).
- [3] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.
- [4] T. Wongpiromsarn, U. Topcu, and R. Murray, "Receding horizon temporal logic planning," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2817–2830, Nov. 2012.
- [5] E. M. Wolff, U. Topcu, and R. M. Murray, "Optimization-based trajectory generation with linear temporal logic specifications," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 5319–5325.
- [6] Z. Kong, A. Jones, and C. Belta, "Temporal logics for learning and detection of anomalous behavior," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1210–1222, Mar. 2017.
- [7] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.
- [8] A. Bhatia, M. R. Maly, L. E. Kavraki, and M. Y. Vardi, "Motion planning with complex goals," *IEEE Robot. Automat. Mag.*, vol. 18, no. 3, pp. 55–64, Sep. 2011.
- [9] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, Feb. 2009, doi: [10.1016/j.automatica.2008.08.008](https://doi.org/10.1016/j.automatica.2008.08.008).
- [10] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis Part I: Quantitative model-based methods," *Comput. & Chem. Eng.*, vol. 27, no. 3, pp. 293–311, Mar. 2003. [Online]. Available: [www.elsevier.com/locate/comchemeng](http://www.elsevier.com/locate/comchemeng)
- [11] F. Djemou, Z. Xu, and U. Topcu, "Probabilistic swarm guidance subject to graph temporal logic specifications," in *Proc. Robot.: Sci. Syst.*, 2020.
- [12] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 293–311, 2003.
- [13] F. Harirchi, S. Z. Yong, and N. Ozay, "Guaranteed fault detection and isolation for switched affine models," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 5161–5167.
- [14] Z. Xu, M. Birtwistle, C. Belta, and A. Julius, "A temporal logic inference approach for model discrimination," *IEEE Life Sci. Lett.*, vol. 2, no. 3, pp. 19–22, Sep. 2016.
- [15] R. S. Smith and J. C. Doyle, "Model invalidation: A connection between robust control and identification," in *Proc. Amer. Control Conf.*, 1989, pp. 1435–1440.
- [16] F. D. Bianchi and R. S. Sánchez-Peña, "Robust identification/invalidation in an LPV framework," *Int. J. Robust Nonlinear Control*, vol. 20, no. 3, pp. 301–312, 2010.
- [17] M. Sznaier and M. C. Mazzaro, "An LMI approach to control-oriented identification and model (in)validation of LPV systems," *IEEE Trans. Autom. Control*, vol. 48, no. 9, pp. 1619–1624, Sep. 2003.
- [18] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.
- [19] Z. Jin, Q. Shen, and S. Z. Yong, "Optimization-based approaches for affine abstraction and model discrimination of uncertain nonlinear systems," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 7976–7981.
- [20] N. Ozay, M. Sznaier, and C. Lagoa, "Convex certificates for model (in)validation of switched affine systems with unknown switches," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 2921–2932, Nov. 2014.
- [21] 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.
- [22] 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*. Berlin, Germany: Springer, 2018, pp. 227–251.
- [23] H. Lou and R. Yang, "Conditions for distinguishability and observability of switched linear systems," *Nonlinear Anal.: Hybrid Syst.*, vol. 5, pp. 427–445, 2011.
- [24] M. Babaali and G. Pappas, "Observability of switched linear systems in continuous time," in *Proc. Int. Workshop Hybrid Syst.: Comput. Control*, 2005, pp. 103–117.
- [25] E. Santis and M. Benedetto, "Observability and diagnosability of finite state systems: A unifying framework," *Automatica*, vol. 81, pp. 115–122, 2017.
- [26] N. Ramdani, L. Trave-Massuyes, and C. Jaubertie, "Mode discernibility and bounded-error state estimation for nonlinear hybrid systems," *Automatica*, vol. 91, pp. 118–125, 2018.
- [27] S. Jiang and R. Kumar, "Failure diagnosis of discrete-event systems with linear-time temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 49, no. 6, pp. 934–945, Jun. 2004.
- [28] L. Yang and N. Ozay, "Fault detectability analysis of switched affine systems with linear temporal logic constraints," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 5779–5786.
- [29] R. Niu, S. M. Hassaan, L. Yang, Z. Jin, and S. Z. Yong, "Model discrimination of switched nonlinear systems with temporal logic-constrained switching," *IEEE Control Syst. Lett.*, vol. 6, pp. 151–156, 2022.
- [30] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer, 2006.
- [31] M. Milanese and C. Novara, "Set membership identification of nonlinear systems," *Automatica*, vol. 40, pp. 957–975, 2004.
- [32] M. Canale, L. Fagiano, and M. Signorile, "Nonlinear model predictive control from data: A set membership approach," *Int. J. Robust Nonlinear Control*, vol. 24, no. 1, pp. 123–139, 2014.

- [33] Z. Zabinsky, R. Smith, and B. Kristinsdottir, "Optimal estimation of univariate black-box Lipschitz functions with upper and lower error bounds," *Comput. Operations Res.*, vol. 30, no. 10, pp. 1539–1553, 2003.
- [34] G. Beliakov, "Interpolation of Lipschitz functions," *J. Comput. Appl. Math.*, vol. 196, no. 1, pp. 20–44, 2006.
- [35] Z. Jin, M. Khajenejad, and S. Yong, "Data-driven model invalidation for unknown Lipschitz continuous systems via abstraction," in *Proc. Amer. Control Conf.*, 2020, pp. 2975–2980.
- [36] Z. Jin, M. Khajenejad, and S. Z. Yong, "Data-driven abstraction and model invalidation for unknown systems with bounded Jacobians," *IEEE Control Syst. Lett.*, vol. 6, pp. 3421–3426, 2022.
- [37] J. Calliess, "Conservative decision-making and inference in uncertain dynamical systems," Ph.D. dissertation, University of Oxford, Oxford, U.K., 2014.
- [38] J. Manzano, D. d. l. Pena, J.-P. Calliess, and D. Limon, "Component-wise hölder inference for robust learning-based MPC," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5577–5583, Nov. 2021.
- [39] N. Baharisangari, J.-R. Gaglione, D. Neider, U. Topcu, and Z. Xu, "Uncertainty-aware signal temporal logic inference," in *Proc. Int. Workshop Numer. Softw. Verification.*, 2022, pp. 61–85.
- [40] R. Roy, J.-R. Gaglione, N. Baharisangari, D. Neider, Z. Xu, and U. Topcu, "Learning interpretable temporal properties from positive examples only," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2023, pp. 6507–6515.
- [41] J.-R. Gaglione, D. Neider, R. Roy, U. Topcu, and Z. Xu, "Learning linear temporal properties from noisy data: A MaxSAT-based approach," in *Proc. 19th Int. Symp. Automated Technol. Verification Anal.*, 2021, pp. 74–90.
- [42] Z. Xu and A. A. Julius, "Robust temporal logic inference for provably correct fault detection and privacy preservation of switched systems," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3010–3021, Sep. 2019.
- [43] Z. Xu, S. Saha, B. Hu, S. Mishra, and A. A. Julius, "Advisory temporal logic inference and controller design for semiautonomous robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 1, pp. 459–477, Jan. 2019.
- [44] Z. Xu, C. Belta, and A. Julius, "Temporal logic inference with prior information: An application to robot arm movements," in *Proc. IFAC-Papers OnLine*, vol. 48, no. 27, 2015, pp. 141–146.
- [45] Z. Xu, M. Ornik, A. A. Julius, and U. Topcu, "Information-guided temporal logic inference with prior knowledge," in *Proc. Amer. Control Conf.*, 2019, pp. 1891–1897.
- [46] Z. Xu, A. J. Nettekoven, A. A. Julius, and U. Topcu, "Graph temporal logic inference for classification and identification," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 4761–4768.
- [47] C. Yoo and C. Belta, "Rich time series classification using temporal logic," in *Proc. Robot. Sci. Syst.*, 2017.
- [48] C. Lemieux, D. Park, and I. Beschastnikh, "General LTL specification mining (T)," in *Proc. 30th IEEE/ACM Int. Conf. Automated Softw. Eng.*, 2015, pp. 81–92.
- [49] A. Kreutzmann, I. Colonius, D. Wolter, F. Dylla, L. Frommberger, and C. Freksa, "Temporal logic for process specification and recognition," *Intell. Service Robot.*, vol. 6, pp. 5–18, 2013.
- [50] R. Yan, Z. Xu, and A. Julius, "Swarm signal temporal logic inference for swarm behavior analysis," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 3021–3028, Jul. 2019.
- [51] D. Kasenberg and M. Scheutz, "Interpretable apprenticeship learning with temporal logic specifications," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 4914–4921.
- [52] Z. Xu and A. A. Julius, "Census signal temporal logic inference for multiagent group behavior analysis," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 1, pp. 264–277, Jan. 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/7587357/>
- [53] A. Camacho and S. A. McIlraith, "Learning interpretable models expressed in linear temporal logic," in *Proc. Int. Conf. Automated Plan. Scheduling*, 2021, pp. 621–630. [Online]. Available: <https://ojs.aaai.org/index.php/ICAPS/article/view/3529>
- [54] A. Shah, P. Kamath, S. Li, and J. Shah, "Bayesian inference of temporal task specifications from demonstrations," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3808–3817.
- [55] F. Pigozzi, E. Medvet, and L. Nenzi, "Mining road traffic rules with signal temporal logic and grammar-based genetic programming," *Appl. Sci.*, vol. 11, no. 22, 2021, Art. no. 10573. [Online]. Available: <https://www.mdpi.com/2076-3417/11/22/10573>
- [56] M. Ryoo and J. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1709–1718.
- [57] J. H. Dawes and G. Reger, "Explaining violations of properties in control-flow temporal logic," in *Proc. Int. Conf. Runtime Verification*, 2019, pp. 202–220.
- [58] P. O. Meredith, D. Jin, F. Chen, and G. Rosu, "Efficient monitoring of parametric context-free patterns," in *Proc. IEEE/ACM 23rd Int. Conf. Automated Softw. Eng.*, 2008, pp. 148–157.
- [59] D. Neider and I. Gavran, "Learning linear temporal properties," in *Proc. 18th Conf. Formal Methods Comput.-Aided Des.*, 2018, pp. 1–10.
- [60] K. Schneider, "Temporal logics," in *Verification Reactive Syst.: Formal Methods Algorithms*, 2004, pp. 279–403.
- [61] T. Jiang, M. Li, B. Ravikumar, and K. W. Regan, "Formal grammars and languages," in *Algorithms and Theory of Computation Handbook: General Concepts and Techniques*, London, U.K.: CRC, 2010, pp. 1–20.
- [62] N. Chomsky, "Three models for the description of language," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 113–124, 1956.
- [63] G. Jäger and J. Rogers, "Formal language theory: Refining the Chomsky hierarchy," *Philos. Trans. Roy. Soc. B: Biol. Sci.*, vol. 367, pp. 1956–1970, 2012.
- [64] S. Karaman, R. G. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 2117–2122.
- [65] M. Bhagwat, Z. Jin, and S. Z. Yong, "Computation-aware data-driven model discrimination with application to driver intent identification," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 6848–6854.
- [66] K. Singh, Q. Shen, and S. Z. Yong, "Mesh-based affine abstraction of nonlinear systems with tighter bounds," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 3056–3061.
- [67] M. Kearns, U. Vazirani, and U. Vazirani, *An Introduction to Computational Learning Theory*. Cambridge, MA, USA: MIT Press, 1994.
- [68] Z. Jin, Q. Shen, and S. Z. Yong, "Mesh-based piecewise affine abstraction with polytopic partitions for nonlinear systems," *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1543–1548, Nov. 2021, doi: [10.1109/LC-SYS.2020.3041312](https://doi.org/10.1109/LC-SYS.2020.3041312).
- [69] L. De Moura and N. Björner, "Z3: An efficient SMT solver," in *Proc. Theory Pract. Softw., 14th Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2008, pp. 337–340.
- [70] Gurobi Optimization, Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: <http://www.gurobi.com>
- [71] R. Palm and D. Driankov, "Fuzzy switched hybrid systems-modeling and identification," in *Proc. IEEE Int. Symp. Intell. Control Held Jointly IEEE Int. Symp. Comput. Intell. Robot. Automat.*, 1998, pp. 130–135.
- [72] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, pp. 497–516, 1957.



**ZEYUAN JIN** (Student Member, IEEE) received the B.S. degree in energy engineering from Zhejiang University, Zhejiang, China, in 2016, the M.S. degree in mechanical engineering from Columbia University, New York, NY, USA in 2018, and the Ph.D. degree in mechanical engineering from Arizona State University, Tempe, AZ, USA, in 2023. His current research interests include control systems, motion planning, optimization and machine learning with applications to autonomous and robotics.



**NASIM BAHARISANGARI** received the B.S. degree in mechanical engineering from Guilan University, Rasht, Iran, in 2016, and the M.S. degree in mechanical engineering from Clemson University, Clemson, SC, USA, in 2020. She is currently working toward the Ph.D. degree in mechanical engineering with Arizona State University, Tempe, AZ, USA, with a research focus on formal methods, optimization, machine learning, and control systems.



**ZHE XU** (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Tianjin University, Tianjin, China, in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2018. He is currently an Assistant Professor with the School for Engineering of Matter, Transport, and Energy, Arizona State University (ASU), Tempe, AZ, USA. Before joining ASU, he was a Postdoctoral Researcher with the Oden Institute for Computational Engineering

and Sciences, University of Texas at Austin, Austin, TX, USA. His research interests include formal methods, autonomous systems, control systems, and reinforcement learning. He was the recipient of the ASU Fulton Schools of Engineering Top 5% Teaching Recognition Award in 2023.



**SZE ZHENG YONG** (Member, IEEE) received the Dipl.-Ing.(FH) degree in automotive engineering with a specialization in mechatronics/control systems from the Esslingen University of Applied Sciences, Esslingen am Neckar, Germany, in 2008, and the S.M. and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2010 and 2016, respectively. He is currently an Associate Professor with the Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA, USA. Prior to that, he was an Assistant Professor with the School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, USA. His research interests include control, estimation, planning, learning and analysis of hybrid systems, with applications to autonomous, robotic and cyber-physical dynamic systems and their safety, robustness and resilience. He was the recipient of the DARPA Young Faculty Award in 2018, the NSF CAREER and NASA Early Career Faculty awards in 2020, ONR Young Investigator Program award in 2022, and the George N. Saridis Best Transactions Paper Award in 2023.

and Sciences, University of Texas at Austin, Austin, TX, USA. His research interests include formal methods, autonomous systems, control systems, and reinforcement learning. He was the recipient of the ASU Fulton Schools of Engineering Top 5% Teaching Recognition Award in 2023.