

# Graph Temporal Logic Inference for Classification and Identification

Zhe Xu, Alexander J Nettekoven, A. Agung Julius, Ufuk Topcu \*

March 26, 2019

## Abstract

Inferring *spatial-temporal properties* from data is important for many complex systems, such as additive manufacturing systems, swarm robotic systems and biological networks. Such systems can often be modeled as a labeled graph where labels on the nodes and edges represent relevant measurements such as temperatures and distances. We introduce *graph temporal logic* (GTL) which can express properties such as “when-ever a node’s label is above 10, for the next 3 time units there are always at least two neighboring nodes with an edge label of at most 2 where the node labels are above 5”. This paper is a first attempt to infer spatial (graph) temporal logic formulas from data for classification and identification. For classification, we infer a GTL formula that classifies two sets of *graph temporal trajectories* with minimal misclassification rate. For identification, we infer a GTL formula that is *informative* and is satisfied by the *graph temporal trajectories* in the dataset with high probability. The informativeness of a GTL formula is measured by the information gain with respect to given prior knowledge represented by a prior probability distribution. We implement the proposed approach to classify the graph patterns of tensile specimens built from selective laser sintering (SLS) process with varying strengths, and to identify informative spatial-temporal patterns from experimental data of the SLS cooldown process and simulation data of a swarm of robots.

## 1 Introduction

Inferring *spatial-temporal properties* from data is important in many applications (e.g., additive manufacturing processes, swarm robotics and biological networks). Consider a powder bed of selective laser sintering (i.e., SLS, one type of additive manufacturing) processes [1] modeled as a labeled graph (as shown in Fig. 1). Each subregion of the powder bed is considered a node of the graph and edges exist between nodes (subregions) within certain distance. Given the time-varying temperature labels at each node and the distance

---

\*Zhe Xu is with the Institute for Computational Engineering and Sciences (ICES), University of Texas, Austin, Austin, TX 78712, Alexander Nettekoven is with the Walker Department of Mechanical Engineering at the University of Texas at Austin, Austin, TX 78712, A. Agung Julius is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, Ufuk Topcu is with the Department of Aerospace Engineering and Engineering Mechanics, and the Institute for Computational Engineering and Sciences (ICES), University of Texas, Austin, Austin, TX 78712, e-mail: zhexu@utexas.edu, nettekoven@utexas.edu, juliua2@rpi.edu, utopcu@utexas.edu.

labels on each edge of the graph, we intend to infer knowledge that can characterize the spatial-temporal patterns that emerge in this process.

The representation of the inferred knowledge should be both interpretable to humans and amenable to rigorous mathematical analysis. Variants of temporal logic and spatial logic can express temporal and spatial patterns in a form that resembles natural language [2]. Furthermore, such expressions are suitable for verification and controller synthesis. Over the past decade, there has been a growing interest in inferring temporal logic formulas from system trajectories [2, 3, 4, 5, 6, 7, 8, 9]. However, to the best of our knowledge, there has been no work on inferring spatial or spatial temporal logic formulas from data.

Two different categories exist for inferring such spatial or temporal logics from data: *classification* and *identification*. Given two sets of data, the classification problem is about constructing spatial temporal logic formulas that can classify these two sets of data with minimal misclassification rate. The identification problem is about identifying spatial temporal logic formulas that best *fit* one set of data.

For identification, one measure of the quality of the inferred formula is its *informativeness*, i.e., the extent to which the inferred formula deviates from prior knowledge. In the example as shown in Fig. 1, suppose that we are given two candidate formulas: one reads as “for every node, either it is red or it is not red” and the other one reads as “for every blue node, there exist at least two red nodes among the neighbors of its neighbors with edge labels of at least 2”. While both formulas are consistent with the labeled graph, the first formula is actually a *tautology* and holds for any labeled graph. In comparison, the second formula describes a specific pattern existing in this labeled graph, hence it is considered to be more *informative* than the first formula.

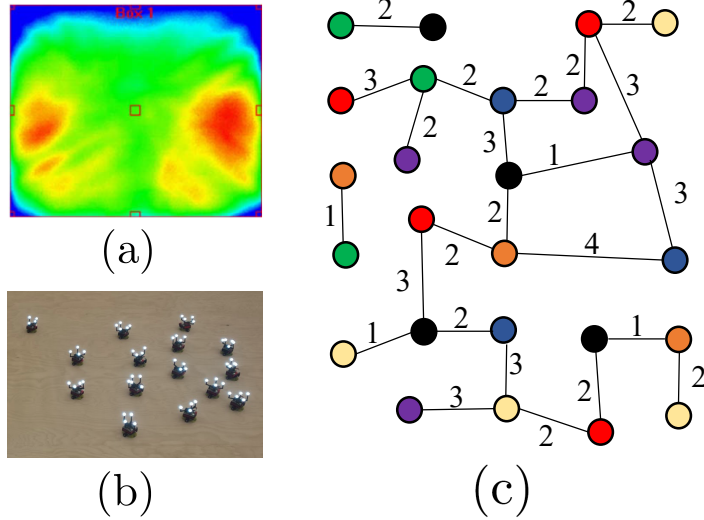


Figure 1: The powder bed of SLS process recorded with an infrared camera (a) [1] and a swarm of mobile robots (b) [10] can be both modeled as a labeled graph (c), where the colors indicate node labels (e.g., temperature, probabilistic density) and the numbers indicate edge labels (e.g., distance).

In this paper, we first introduce *parametric graph temporal logic* (pGTL), which is an

extension of parametric linear temporal logic and focuses on the spatial-temporal properties of the labels on a graph. A pGTL formula has free parameters in the predicates and operators. A *graph temporal logic* (GTL) formula is then induced by a pGTL formula by assigning real values to the parameters of the pGTL formula. We study the following two problems of inferring GTL formulas from spatial-temporal data over a graph:

- *Inferring GTL formulas for classification:*  
We infer a GTL formula that best classifies two sets of *graph-temporal trajectories* (formalized in Sec. 2).
- *Inferring informative GTL formulas for identification:*  
We infer a GTL formula that is consistent with a set of *graph-temporal trajectories* and provides a high information gain (formalized in Sec. 4) over a given prior probability distribution.

In Sec. 5.1, we implement the classification method to infer GTL formulas that can classify the graph patterns of tensile specimens built from selective laser sintering (SLS) process with varying strengths. In Sec. 5.2 and 5.3, we implement the identification method to infer informative GTL formulas from experimental data of the SLS cooldown process and simulation data of a swarm of robots, respectively.

**Related Work.** There exist several spatial (graph) temporal logics in the literature, such as spatial-temporal logic (SpaTeL) in [11] and signal spatio-temporal logic (SSTL) in [12]. Our proposed GTL is different from both SpaTeL and SSTL as GTL focuses on the propositions on the node labels and edge labels of a graph, and whether there exist certain number of neighbors that satisfy the node propositions with the connecting edges satisfying the edge propositions. GTL is also different from the logics of graphs in [13, 14] as they consider logical statements about the structure of the graphs and the changes in the structure, while we consider logical statements about (possibly time-varying) labels that are defined on graphs with fixed structure.

Our approach of inferring GTL formulas from data is closely related to inferring temporal logic formulas from data. The work in [2, 5, 8] focus on inferring temporal logic formulas for classifying two sets of trajectories, while the work in [3, 6, 9, 15] focus on identifying temporal logic formulas from system trajectories.

## 2 Parametric Graph Temporal Logic and Graph Temporal Logic

In this section, we introduce parametric graph temporal logic (pGTL) and graph temporal logic (GTL).

### 2.1 Node and Edge Propositions

Let  $G = (V, E)$  be an undirected graph, where  $V$  is a finite set of nodes and  $E$  is a finite set of edges. We use  $\mathcal{X}$  to denote a (possibly infinite) set of node labels and  $\mathcal{Y}$  to denote a (possibly infinite) set of edge labels. We use  $s(e) = \{v_1, v_2\}$  to denote the fact that the edge  $e \in E$  connects  $v_1 \in V$  and  $v_2 \in V$ .  $\mathbb{T} = \{1, 2, \dots\}$  is a discrete set of time indices. A graph with node labels and edge labels is also called a *labeled graph*.

**Definition 1** A graph-temporal trajectory on a graph  $G$  is a tuple  $g = (x, y)$ , where  $x : V \times \mathbb{T} \rightarrow \mathcal{X}$  assigns a node label for each node  $v \in V$  at each time index  $k \in \mathbb{T}$ , and  $y : E \times \mathbb{T} \rightarrow \mathcal{Y}$  assigns an edge label for each edge  $e \in E$  at each time index  $k \in \mathbb{T}$ .

We use  $x(v, k)$  to denote the label of node  $v$  at time index  $k$  and  $y(e, k)$  to denote the label of edge  $e$  at time index  $k$ .

**Definition 2** An atomic node proposition is a predicate on  $\mathcal{X}$ , i.e., a Boolean valued map from  $\mathcal{X}$ . An edge proposition is a predicate on  $\mathcal{Y}$ .

We use  $\pi$  to denote an atomic node proposition, and  $\mathcal{O}(\pi)$  to denote the subset of  $\mathcal{X}$  for which  $\pi$  is true. We use  $\rho$  to denote an edge proposition, and  $\mathcal{O}(\rho)$  to denote the subset of  $\mathcal{Y}$  for which  $\rho$  is true.

We define that a graph-temporal trajectory  $g = (x, y)$  satisfies an atomic node proposition  $\pi$  at a node  $v$  and at a time index  $k$ , denoted as  $(g, v, k) \models \pi$ , if and only if  $x(v, k) \in \mathcal{O}(\pi)$ . Similarly, we define that a graph-temporal trajectory  $g = (x, y)$  satisfies an edge proposition  $\rho$  at an edge  $e$  and at a time index  $k$ , denoted as  $(g, e, k) \models \rho$ , if and only if  $y(e, k) \in \mathcal{O}(\rho)$ .

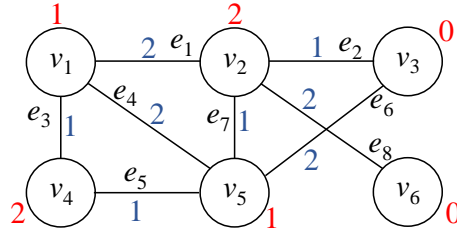


Figure 2: An example of a graph-temporal trajectory on an undirected graph, with the red numbers indicating node labels, and the blue numbers indicating edge labels, all at a fixed time index  $k$ .

**Example 1** For the graph in Fig. 2, the node and edge labels are from a graph-temporal trajectory  $g$  by fixing a time index  $k$ . The atomic node proposition  $\pi = (x \leq 0)$  is satisfied by  $g$  at  $v_3$  and  $v_6$  at time index  $k$ . The edge proposition  $\rho = (y \geq 2)$  is satisfied by  $g$  at  $e_1$ ,  $e_4$ ,  $e_6$  and  $e_8$  at time index  $k$ .

**Definition 3** Given a graph-temporal trajectory  $g = (x, y)$  on a graph  $G$ , a subset  $V' \subseteq V$  of nodes and an edge proposition  $\rho$ , we define the neighbor operation  $\bigcirc_\rho : 2^V \times \mathbb{T} \rightarrow 2^V \times \mathbb{T}$  as

$$\begin{aligned} \bigcirc_\rho(V', k) = \\ (\{v \mid \exists v' \in V', \exists e \in E, s(e) = \{v', v\}, (g, e, k) \models \rho\}, k). \end{aligned}$$

Intuitively,  $\bigcirc_\rho(V', k)$  consists of nodes that can be reached from  $V'$  through an edge where the edge proposition  $\rho$  is satisfied by  $g$  at time index  $k$ . Note that neighbor operations can be applied successively.

**Example 2** For the graph-temporal trajectory  $g$  on the graph  $G$  at time index  $k$  in Fig. 2,

$$\begin{aligned} \bigcirc_{y \leq 1}(\{v_4\}, k) &= (\{v_1, v_5\}, k), \\ \bigcirc_{y \leq 1} \bigcirc_{y \leq 1}(\{v_4\}, k) &= \bigcirc_{y \leq 1}(\{v_1, v_5\}, k) = (\{v_2, v_4\}, k). \end{aligned}$$

## 2.2 pGTL Formulas and GTL formulas

We define the syntax of a parametric graph temporal logic (pGTL) formula  $\varphi$  recursively as

$$\varphi := \pi \mid \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathcal{U} \varphi \mid \Diamond_{\sim i} \varphi,$$

where  $n$  and  $N$  are positive integers,  $\pi$  is an atomic node proposition,  $\rho_i$  ( $i = 1, \dots, n$ ) are edge propositions,  $\exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi$  reads as “there exists at least  $N$  nodes under the neighbor operation  $\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1}$  that satisfy  $\varphi$ ”,  $\neg$  and  $\wedge$  stand for negation and conjunction respectively,  $\mathcal{U}$  is a temporal operator representing “until”,  $\Diamond_{\sim i}$  is a parametrized temporal operator representing “parametrized eventually”, where  $\sim \in \{\geq, \leq\}$  and  $i \in \mathbb{T}$  is a temporal parameter. We can also derive  $\vee$  (disjunction),  $\Diamond$  (eventually),  $\Box$  (always),  $\Box_{\sim i}$  (parametrized always),  $\mathcal{U}_{\sim i}$  (parametrized until) and  $\Rightarrow$  (implication) from the above-mentioned operators [16]. We can also derive the parametrized temporal operators  $\Diamond_{\geq i_1, \leq i_2}$  and  $\Box_{\geq i_1, \leq i_2}$  ( $i_1 < i_2, i_1, i_2 \in \mathbb{T}$ ) as  $\Diamond_{\geq i_1, \leq i_2} \phi = \Diamond_{\geq i_1} \phi \wedge \Diamond_{\leq i_2} \phi$  and  $\Box_{\geq i_1, \leq i_2} \phi = \Box_{\geq i_1} \phi \wedge \Box_{\leq i_2} \phi$ .

The satisfaction relation  $(g, v, k) \models \varphi$  for a graph-temporal trajectory  $g$  at node  $v$  at time index  $k$  with respect to a pGTL formula  $\varphi$  is defined recursively as follows:

$$\begin{aligned} (g, v, k) \models \pi & \quad \text{iff} \quad x(v, k) \in \mathcal{O}(\pi) \\ (g, v, k) \models \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi & \quad \text{iff} \quad \exists v_1, \dots, v_N \ (v_i \neq v_j \text{ for } i \neq j), \text{ s.t.} \\ & \quad \forall i, (v_i, k) \in \bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1}(v, k) \text{ and } (g, v_i, k) \models \varphi \\ (g, v, k) \models \neg\varphi & \quad \text{iff} \quad (g, v, k) \not\models \varphi \\ (g, v, k) \models \varphi_1 \wedge \varphi_2 & \quad \text{iff} \quad (g, v, k) \models \varphi_1 \text{ and } (g, v, k) \models \varphi_2 \\ (g, v, k) \models \varphi_1 \mathcal{U} \varphi_2 & \quad \text{iff} \quad \exists k' \geq k, \text{ s.t. } (g, v, k') \models \varphi_2, \\ & \quad (g, v, k'') \models \varphi_1, \forall k'' \in [k, k'] \\ (g, v, k) \models \Diamond_{\sim i} \varphi & \quad \text{iff} \quad \exists k' \sim k + i, \text{ s.t. } (g, v, k') \models \varphi \end{aligned}$$

Intuitively,  $\exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi$  is satisfied by a graph-temporal trajectory  $g$  at a node  $v \in V$  and at a time index  $k$  if there exist at least  $N$  nodes in  $(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})(v, k)$  where  $\varphi$  is satisfied by  $g$  at time index  $k$ . Note that, by definition, if  $(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})(v, k)$  consists of fewer than  $N$  nodes, then  $\exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi$  is false.

We also define that a graph-temporal trajectory  $g$  satisfies a pGTL formula  $\varphi$  at a node  $v$ , denoted as  $(g, v) \models \varphi$ , if  $g$  satisfies  $\varphi$  at node  $v$  at time index 1.

**Definition 4** We define a graph temporal logic (GTL) formula  $\varphi_\theta$  as a pGTL formula  $\varphi$  with fixed parameter valuation  $\theta$ .

**Example 3** For the pGTL formula  $\varphi = \exists^N \bigcirc_{y \leq a} (x \geq b)$ , we can induce a GTL formula  $\varphi_\theta = \exists^2 \bigcirc_{y \leq 1} (x \geq 1)$  with  $\theta([N, a, b]) = [2, 1, 1]$ . For the graph-temporal trajectory  $g$  on the graph shown in Fig. 2, the set of nodes where  $\varphi_\theta$  is satisfied by  $g$  at time index  $k$  are  $\{v_4, v_5\}$ .

## 2.3 Subtypes of pGTL and GTL Formulas

Graph-temporal trajectories of finite time length are sufficient to satisfy (resp. violate) syntactically co-safe (resp. safe) pGTL formulas, which are defined as follows.

**Definition 5** The syntax of the syntactically co-safe pGTL formula is defined recursively as

$$\begin{aligned} \varphi := & \top \mid \pi \mid \neg\pi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Diamond\varphi \mid \varphi_1 \mathcal{U}\varphi_2 \mid \Diamond_{\sim i}\varphi \\ & \mid \Box_{\leq i}\varphi \mid \varphi_1 \mathcal{U}_{\sim i}\varphi_2 \mid \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi. \end{aligned}$$

**Definition 6** The syntax of the syntactically safe pGTL formula is defined as

$$\begin{aligned} \varphi := & \perp \mid \pi \mid \neg\pi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box\varphi \mid \Diamond_{\leq i}\varphi \mid \Box_{\sim i}\varphi \\ & \mid \varphi_1 \mathcal{U}_{\leq i}\varphi_2 \mid \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\varphi. \end{aligned}$$

We further introduce type-I and type-II pGTL formulas.

(1) A type-I pGTL formula  $\varphi$  is defined as

$$\varphi := \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\pi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathcal{U}\varphi \mid \Diamond_{\sim i}\varphi.$$

(2) A type-II pGTL formula  $\varphi$  is defined recursively as

$$\varphi := \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\phi,$$

where  $\phi$  is defined recursively as

$$\phi := \pi \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathcal{U}\phi \mid \Diamond_{\sim i}\phi.$$

The subtypes of GTL Formulas can be defined similarly with fixed parameter valuations. A pGTL formula can be neither a type-I pGTL formula nor a type-II pGTL formula. A pGTL formula could also be of more than one subtypes. For example, if a pGTL formula is both a type-I pGTL formula and a syntactically co-safe pGTL formula, then it is called a type-I syntactically co-safe pGTL formula.

**Definition 7** A deterministic finite automaton (DFA) is a tuple  $\mathcal{A} = (\mathcal{Q}, q_0, \Sigma, \delta, Acc)$  where  $\mathcal{Q} = \{q_0, q_1, \dots, q_{K-1}\}$  is a finite set of states,  $q_0$  is the initial state,  $\Sigma$  is the alphabet,  $\delta : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$  is the transition relation and  $Acc \subseteq 2^{\mathcal{Q}}$  is a finite set of accepting states [17].

We use  $AP^I$  and  $AP^{II}$  to denote the sets of *atomic predicates* in the form of  $\exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\pi$  and  $\pi$ , respectively. At a node  $v$  and over  $AP^I$  (resp.  $AP^{II}$ ), the *word* generated by a graph-temporal trajectory  $g = (x, y)$  is a sequence  $\mathcal{L}^v(x(\cdot, 1), y(\cdot, 1)), \dots, \mathcal{L}^v(x(\cdot, L), y(\cdot, L))$ , where  $\mathcal{L}^v : \mathcal{X} \times \mathcal{Y} \rightarrow 2^{AP^I}$  (resp.  $2^{AP^{II}}$ ) is a labeling function assigning a subset of atomic predicates in  $AP^I$  (resp.  $AP^{II}$ ) that hold true at node  $v$  to each  $(x(\cdot, k), y(\cdot, k))$ ,  $k \in [1, L]$ . If a pGTL formula  $\varphi$  is a type-I pGTL formula and it is syntactically co-safe (resp. safe), then we can build a DFA  $\mathcal{A}^{\varphi_\theta, v}$  (resp.  $\mathcal{A}^{\neg\varphi_\theta, v}$ ) over  $AP^I$  that accepts precisely the words generated by graph-temporal trajectories that satisfy (resp. violate) the GTL formula  $\varphi_\theta$  at node  $v$  for any  $\theta$ . If a pGTL formula  $\varphi = \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\phi$  is a type-II pGTL formula and it is syntactically co-safe (resp. safe), we can build a DFA  $\mathcal{A}^{\phi_\theta, v}$  (resp.  $\mathcal{A}^{\neg\phi_\theta, v}$ ) over  $AP^{II}$  that accepts precisely the words generated by graph-temporal trajectories that satisfy (resp. violate)  $\phi_{\theta'}$  at node  $v$  for any  $\theta'$  [17].

In the following sections of the paper, we only focus on type I/type II syntactically co-safe/syntactically safe pGTL (GTL) formulas and we simply call them pGTL (GTL) formulas for conciseness. We define that a syntactically co-safe GTL formula  $\varphi_\theta$  is violated by a graph-temporal trajectory  $g$  of finite time length  $L$  at a node  $v$  if  $\varphi_\theta$  is not satisfied by  $g$  at node  $v$ ; and a syntactically safe GTL formula  $\varphi_\theta$  is satisfied by a graph-temporal trajectory  $g$  of finite time length  $L$  at a node  $v$  if  $\varphi_\theta$  is not violated by  $g$  at node  $v$ .

### 3 Graph Temporal Logic Inference for Classification

In this section, we present the problem formulation and solution to infer GTL formulas for classification.

Suppose that we are given a set  $\mathcal{D}_L^G = \{(g_k, l_k)\}_{k=1}^{N_D}$  of *labeled* graph-temporal trajectories of time length  $L$  on a graph  $G$ , where the *classification labels*  $l_k = 1$  and  $l_k = -1$  represent desired and undesired behaviors, respectively. For a GTL formula  $\varphi_\theta$ , we define the *satisfaction signature*  $\zeta_{\varphi_\theta}(g_k, v)$  of a graph-temporal trajectory  $g_k$  at node  $v$  as follows:  $\zeta_{\varphi_\theta}(g_k, v) = 1$  if  $(g_k, v) \models \varphi_\theta$ ; and  $\zeta_{\varphi_\theta}(g_k, v) = -1$  if  $(g_k, v) \not\models \varphi_\theta$ . A labeled graph-temporal trajectory  $(g_k, l_k)$  is misclassified by  $\varphi_\theta$  at node  $v$  if  $\zeta_{\varphi_\theta}(g_k, v) \neq l_k$ .

We define the nodal misclassification rate of  $\varphi_\theta$  in  $\mathcal{D}_L^G$  as

$$MR(\mathcal{D}_L^G, \varphi_\theta) = \frac{\sum_{v \in V} |\{(g_k, l_k) \in \mathcal{D}_L^G : \zeta_{\varphi_\theta}(g_k, v) \neq l_k\}|}{|\mathcal{D}_L^G| |V|},$$

where  $|S|$  denotes the cardinality of a set  $S$ .

The *size* of a GTL formula  $\varphi_\theta$ , denoted as  $\eta(\varphi_\theta)$ , is defined as the number of *Boolean connectives* (i.e., conjunctions or disjunctions) in  $\varphi_\theta$ . Note that logically equivalent formulas may have different sizes.

**Problem 1** *Given a dataset  $\mathcal{D}_L^G = \{g_1, \dots, g_m\}$ , a prior probability distribution  $\mathcal{F}_L^G$ , a real constant  $m_{\text{th}} \in [0, 1)$  and an integer constant  $\eta_{\text{th}} \in (0, \infty)$ , construct a GTL formula  $\varphi_\theta$  that satisfies the following two constraints:*

- *classification constraint:  $MR(\mathcal{D}_L^G, \varphi_\theta) \leq m_{\text{th}}$ , i.e., the nodal misclassification rate should not exceed  $m_{\text{th}}$ ;*
- *size constraint:  $\eta(\varphi_\theta) \leq \eta_{\text{th}}$ , i.e., the size of  $\varphi_\theta$  should not exceed  $\eta_{\text{th}}$ .*

Most existing approaches for inferring temporal logic formulas for classification apply readily to solve Problem 1. As an example, we use the *pruning and growing* approach illustrated in [2]. We start from a set  $\mathcal{P}$  of *primitive* pGTL formulas (also called *templates*), i.e., pGTL formulas that do not contain conjunctions or disjunctions. We use particle swarm optimization (PSO) [18] to compute the parameter valuation  $\theta$  for each primitive pGTL formula from  $\mathcal{P}$  that minimizes the nodal misclassification rate  $MR(\mathcal{D}_L^G, \varphi_\theta)$ . Other global optimization methods such as simulated annealing [2] and Monte-carlo sampling [19] are also valid candidates for such computations. If a GTL formula that satisfies the classification constraint is not found, we only keep the pGTL formulas in  $\mathcal{P}$  such that the misclassification rates can be achieved below a threshold  $\hat{m}_{\text{th}} \in (m_{\text{th}}, 1)$  (*pruning*). Then we infer a GTL formula  $\varphi_\theta$  in the form of  $\varphi_\theta = \varphi_{\theta_1}^1 \vee \varphi_{\theta_2}^2$  or  $\varphi_\theta = \varphi_{\theta_1}^1 \wedge \varphi_{\theta_2}^2$  (*growing*), where  $\varphi^1$  and  $\varphi^2$  are chosen from the pGTL formulas kept in the first step. In this way, we keep increasing the number of primitive pGTL formulas connected with conjunctions or disjunctions until a GTL formula that satisfies the classification constraint is found, or the size constraint is violated.

### 4 Graph Temporal Logic Inference for Identification

In this section, we present the problem formulation and solution to identify GTL formulas from data.

## 4.1 Problem Formulation

We use  $\mathcal{B}_L^G$  to denote the set of all possible graph-temporal trajectories with time-length  $L$  on the graph  $G$ . We are given a dataset  $\mathcal{S}_L^G = \{g_1, \dots, g_m\} \subset \mathcal{B}_L^G$  as a collection of graph-temporal trajectories. We use  $\mathcal{F}_L^G : \mathcal{B}_L^G \rightarrow [0, 1]$  to denote a prior probability distribution over  $\mathcal{B}_L^G$ , and  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v}$  to denote the probability of a GTL formula  $\varphi_\theta$  being satisfied at node  $v$  based on  $\mathcal{F}_L^G$ , i.e.,

$$\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} = \mathbb{P}\{(g, v) \models \varphi_\theta\}, g \sim \mathcal{F}_L^G.$$

**Assumption 1** We assume that every graph-temporal trajectory in  $\mathcal{B}_L^G$  occurs with non-zero probability based on  $\mathcal{F}_L^G$ .

From Assumption 1, for any  $g \in \mathcal{B}_L^G$ , if  $(g, v) \models \varphi_\theta$ , then  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} > 0$ .

**Definition 8** Given a prior probability distribution  $\mathcal{F}_L^G$ , we define  $\bar{\mathcal{F}}_L^{\varphi_\theta, v} : \mathcal{B}_L^G \rightarrow [0, 1]$  as the posterior probability distribution given that the GTL formula  $\varphi_\theta$  evaluates to true at node  $v$ , which is expressed as

$$\bar{\mathcal{F}}_L^{\varphi_\theta, v}(g) = \begin{cases} \frac{\mathcal{F}_L^G(g)}{\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v}} & \text{if } (g, v) \models \varphi_\theta, \\ 0 & \text{if } (g, v) \not\models \varphi_\theta. \end{cases}$$

**Remark 1** The expression of  $\bar{\mathcal{F}}_L^{\varphi_\theta, v}$  can be directly derived using Bayes' theorem.

**Definition 9** We define

$$\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) := \frac{1}{L} \cdot D_{\text{KL}}(\bar{\mathcal{F}}_L^{\varphi_\theta, v} \| \mathcal{F}_L^G)$$

as the information gain when the prior probability distribution  $\mathcal{F}_L^G$  is updated to the posterior probability distribution  $\bar{\mathcal{F}}_L^{\varphi_\theta, v}$ , where  $D_{\text{KL}}(\bar{\mathcal{F}}_L^{\varphi_\theta, v} \| \mathcal{F}_L^G)$  is the Kullback-Leibler divergence from  $\mathcal{F}_L^G$  to  $\bar{\mathcal{F}}_L^{\varphi_\theta, v}$ .

**Remark 2** If  $\varphi_\theta = \top$ , then obviously  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} = 1$  and  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) = 0$  for any node  $v$ , i.e., tautologies provide no information gain. For completeness, we also define that the information gain  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) = 0$  for any node  $v$  if  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} = 0$ . So if  $\varphi_\theta = \perp$ , then  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} = 0$  and  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) = 0$  for any node  $v$ , i.e., contradictions provide no information gain.

**Definition 10** We define  $\chi(\varphi_\theta, \mathcal{S}_L^G)$  as the averaged proportion of nodes (in  $G$ ) at which the GTL formula  $\varphi_\theta$  is satisfied in the dataset  $\mathcal{S}_L^G$ , i.e.,

$$\chi(\varphi_\theta, \mathcal{S}_L^G) = \frac{\sum_i |\{v \in V \mid (g_i, v) \models \varphi_\theta\}|}{|V| |\mathcal{S}_L^G|}.$$

**Problem 2** Given a dataset  $\mathcal{S}_L^G$ , a prior probability distribution  $\mathcal{F}_L^G$  and a real constant  $p_{\text{th}} \in (0, 1]$ , compute the parameter valuation  $\theta$  for a pGTL formula  $\varphi$  (selected from a set  $\mathcal{P}$  of templates) that maximizes the average information gain at each node  $\frac{1}{|V|} \sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v})$  while satisfying the coverage constraint:  $\chi(\varphi_\theta, \mathcal{S}_L^G) \geq p_{\text{th}}$ , i.e.,  $\varphi_\theta$  is satisfied for at least  $p_{\text{th}}$  proportion of nodes in average in  $\mathcal{S}_L^G$ .



Note that we focus on the parameter identification problem based on pGTL formulas selected from a set of templates and thus the size constraint is not needed.

To solve Problem 2, it is computationally inefficient to use optimization algorithms such as PSO and compute the average information gain for each candidate GTL formula (see Sec. 4.2 for the time complexity). In Sec. 4.3, we prove that the optimal parameter valuation of a GTL formula lies in the *minimal satisfying set* of parameter valuations. Thus we only need to compute the average information gain for the GTL formulas with parameter valuation in the (approximated) minimal satisfying set (see Sec. 4.4).

## 4.2 Computation of Information Gain of GTL Formulas

In this subsection, we present the algorithm for computing the average information gain for GTL formulas.

**Proposition 1** *For a GTL formula  $\varphi_\theta$ , if  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} > 0$ ,*

$$\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) = \frac{-\log \mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v}}{L}.$$

**Proof 1** *Straightforward from Definitions 8 and 9.*

In the following, for computational efficiency we choose prior probability distributions such that there exist no spatial or temporal dependencies for the labels on different nodes and edges. Note that the proposed methodology readily applies to cases where such spatial or temporal dependencies exist, e.g., when the prior distribution is governed by Markov random fields (spatial dependence) or discrete-time Markov chains (temporal dependence), but the computational complexity is significantly increased (see Sec. IV-B of [15] for an example for parametric linear temporal logic).

Algorithm 1 is for computing the average information gain for GTL formulas. We first explain how to compute the average information gain for a type-I syntactically co-safe GTL formula  $\varphi_\theta$ . We use  $p_L^{\varphi_\theta, v}(\ell, q_k)$  to denote the probability of a graph-temporal trajectory of time length  $L$  satisfying  $\varphi_\theta$  at node  $v$ , conditioned on the fact that the state of the DFA  $\mathcal{A}^{\varphi_\theta, v} = (\mathcal{Q}^{\varphi_\theta, v}, q_0^{\varphi_\theta, v}, 2^{AP^1}, \delta^{\varphi_\theta, v}, Acc^{\varphi_\theta, v})$  at time index  $\ell$  ( $1 \leq \ell \leq L$ ) being the state  $q_k$ . We first initialize  $p_L^{\varphi_\theta, v}(L, q_k)$  as (Line 9)

$$p_L^{\varphi_\theta, v}(L, q_k) = \begin{cases} 1 & \text{if } q_k \in Acc^{\varphi_\theta, v}; \\ 0 & \text{otherwise.} \end{cases}$$

We can compute  $p_L^{\varphi_\theta, v}(\ell, q_k)$  recursively as (Line 11)

$$\begin{bmatrix} p_L^{\varphi_\theta, v}(\ell - 1, q_0) \\ \vdots \\ p_L^{\varphi_\theta, v}(\ell - 1, q_K) \end{bmatrix} = \begin{bmatrix} c_{0,0}^\ell & \cdots & c_{0,K}^\ell \\ \vdots & \vdots & \vdots \\ c_{K,0}^\ell & \cdots & c_{K,K}^\ell \end{bmatrix} \begin{bmatrix} p_L^{\varphi_\theta, v}(\ell, q_0) \\ \vdots \\ p_L^{\varphi_\theta, v}(\ell, q_K) \end{bmatrix},$$

where  $c_{j,k}^\ell$  is the probability of transitioning from  $q_j$  to  $q_k$  at time index  $\ell$  and  $c_{j,k}^\ell$  can be calculated based on  $\mathcal{F}_L^G$  (Line 11). Finally, we have  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} = p_L^{\varphi_\theta, v}(1, q_0)$  (Line 12).

For type-I syntactically safe GTL formulas, by replacing each  $\varphi_\theta$  in the above deductions with  $\neg\varphi_\theta$ , we can compute  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} = 1 - p_L^{\neg\varphi_\theta, v}(1, q_0)$  (Line 13).

For a type-II GTL formula  $\varphi_\theta = \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\phi_{\theta'}$ ,  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v}$  can be computed as (Line 17)

$$\sum_{k=N}^{N_{\varphi_\theta}^v} \frac{N_{\varphi_\theta}^v!}{k!(N_{\varphi_\theta}^v - k)!} \mathbb{P}_{\mathcal{F}_L^G, \phi_{\theta'}, v}^k (1 - \mathbb{P}_{\mathcal{F}_L^G, \phi_{\theta'}, v})^{N_{\varphi_\theta}^v - k},$$

where  $N_{\varphi_\theta}^v = |\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1}(v, 1)|$  is the number of nodes in the set  $\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1}(v, 1)$ , and  $\mathbb{P}_{\mathcal{F}_L^G, \phi_{\theta'}, v}$  can be computed in a similar way for type-I GTL formulas.

With  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v}$ , we can compute  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v})$  according to Proposition 1 (Line 18).

The time complexity of Algorithm 1 is  $O(|V|LK^2)$ , where  $|V|$  is the number of nodes in the graph,  $L$  is the time length of graph-temporal trajectories,  $K$  is the number of states of  $\mathcal{A}^{\varphi_\theta, v}$  (for type-I formulas) or  $\mathcal{A}^{\phi_{\theta'}, v}$  (for type-II formulas).

---

**Algorithm 1:** Average information gain computation for GTL formulas.

---

```

1 function ComputeIG( $\mathcal{S}_L^G = \{g_1, \dots, g_m\}, \varphi_\theta, \mathcal{F}_L^G$ )
2   if  $\varphi_\theta$  is a type-I GTL formula then
3      $\psi \leftarrow \varphi_\theta$ 
4   else if  $\varphi_\theta$  is a type-II GTL formula then
5     For  $\varphi_\theta = \exists^N(\bigcirc_{\rho_n} \cdots \bigcirc_{\rho_1})\phi_{\theta'}$ ,  $\psi \leftarrow \phi_{\theta'}$ 
6   for  $v \in V$  do
7     Obtain the DFA  $\mathcal{A}^{\psi, v}$  (resp.  $\mathcal{A}^{\neg\psi, v}$ ) if  $\psi$  is syntactically co-safe (resp.
       syntactically safe)
8     for  $k = 0$  to  $K$  do
9       Initialize  $p_L^{\psi, v}(L, q_k)$ 
10    for  $\ell = L$  to  $2$ ,  $j = 0$  to  $K$  do
11      For each  $k \in [0, K]$ , calculate  $c_{j,k}^\ell p_L^{\psi, v}(\ell - 1, q_j) \leftarrow \sum_{k=0}^K c_{j,k}^\ell p_L^{\psi, v}(\ell, q_k)$ 
12       $\beta \leftarrow p_L^{\psi, v}(1, q_0)$  if  $\psi$  is syntactically co-safe
13       $\beta \leftarrow 1 - p_L^{\psi, v}(1, q_0)$  if  $\psi$  is syntactically safe
14      if  $\psi$  is a type-I GTL formula then
15         $\gamma_v \leftarrow \beta$ 
16      else if  $\psi$  is a type-II GTL formula then
17         $\gamma_v \leftarrow \sum_{k=N}^{N_\psi^v} \frac{N_\psi^v!}{k!(N_\psi^v - k)!} \beta^k (1 - \beta)^{N_\psi^v - k}$ 
18  return  $\mathcal{I} = -\frac{1}{|V|L} \sum_{v \in V} \log(\gamma_v)$ 

```

---

### 4.3 Minimal Satisfying Set of Parameter Valuations

In this subsection, we introduce some related definitions and lemmas, leading to the result in Proposition 2.

**Definition 11** The polarity  $\varrho(p, \varphi)$  of a scalar parameter  $p$  for a pGTL formula  $\varphi$  is defined

recursively as

$$\begin{aligned}
\varrho(p, \neg\varphi) &= \sim \varrho(p, \varphi), \quad \varrho(p, f(x) \leq p) = +, \quad \varrho(p, f(x) \geq p) = -, \\
\varrho(p, \Diamond_{\leq p}\varphi) &= + \circ \varrho(p, \varphi), \quad \varrho(p, \Diamond_{\geq p}\varphi) = - \circ \varrho(p, \varphi), \\
\varrho(p, \varphi \mathcal{M} \psi) &= \varrho(p, \varphi \wedge \psi) = \varrho(p, \varphi) \circ \varrho(p, \psi), \\
\varrho(p, \exists^N \bigcirc_{y \leq p} \varphi) &= + \circ \varrho(p, \varphi), \\
\varrho(p, \exists^N \bigcirc_{y \geq p} \varphi) &= - \circ \varrho(p, \varphi), \\
\varrho(p, \exists^p \bigcirc_\rho \varphi) &= - \circ \varrho(p, \rho) \circ \varrho(p, \varphi), \\
\varrho(p, \varphi) &= U, \text{ iff } p \text{ does not appear in } \varphi,
\end{aligned}$$

where  $f$  is some real-valued function, the operations  $\sim$  and  $\circ$  are as defined in the following table [3]

	$\sim$	$\circ$	$U$	$+$	$-$	$M$
$U$	$U$	$U$	$U$	$+$	$-$	$M$
$+$	$-$	$+$	$+$	$+$	$M$	$M$
$-$	$+$	$-$	$-$	$M$	$-$	$M$
$M$	$M$	$M$	$M$	$M$	$M$	$M$

In this table,  $U, +, -$  and  $M$  represent undefined, positive, negative and mixed polarities respectively.

Intuitively, the polarity  $\varrho(p, \varphi)$  of a scalar parameter  $p$  for a pGTL formula  $\varphi$  is positive, if the pGTL formula  $\varphi$  is easier to be satisfied when  $p$  is increased; and it is negative, if it is easier to be satisfied when  $p$  is decreased.

**Definition 12** For a graph  $G$ , we say that  $\theta$  dominates  $\theta'$  with respect to a pGTL formula  $\varphi$ , denoted as  $\theta \prec_\varphi \theta'$ , if and only if  $\varphi_\theta \Rightarrow \varphi_{\theta'}$  holds true and  $\varphi_{\theta'} \Rightarrow \varphi_\theta$  does not hold true at any node  $v$  for GTL formulas  $\varphi_\theta$  and  $\varphi_{\theta'}$ .

For example, for the six GTL formulas induced from the same pGTL formula  $\varphi$  in Fig. 3, we have  $\varphi_{\theta_2} \prec \varphi_{\theta_1}$ ,  $\varphi_{\theta_3} \prec \varphi_{\theta_2}$ ,  $\varphi_{\theta_4} \prec \varphi_{\theta_2}$ ,  $\varphi_{\theta_5} \prec \varphi_{\theta_2}$ ,  $\varphi_{\theta_6} \prec \varphi_{\theta_3}$  and  $\varphi_{\theta_6} \prec \varphi_{\theta_4}$ .

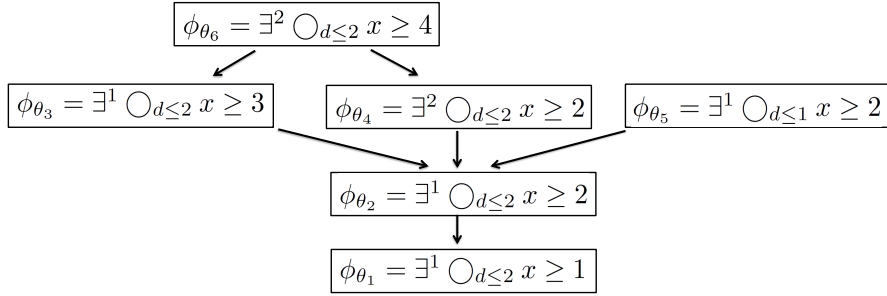


Figure 3: The *domination* relationships among six GTL formulas.

**Lemma 1** For two parameter valuations  $\theta = [\theta_1, \dots, \theta_z]$  and  $\theta' = [\theta'_1, \dots, \theta'_z]$ ,  $\theta \prec_\varphi \theta'$  if  $\theta \neq \theta'$  and the followings hold for each  $i \in [1, z]$ :

$$\begin{cases} \theta_i \leq \theta'_i, & \text{if } \varrho(\theta_i, \varphi) = +; \\ \theta_i \geq \theta'_i, & \text{if } \varrho(\theta_i, \varphi) = -. \end{cases}$$

**Proof 2** Straightforward from Definitions 11 and 12.

**Lemma 2** For two parameter valuations  $\theta$  and  $\theta'$ , if  $\theta \prec_\varphi \theta'$  for a pGTL formula  $\varphi$ , then  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) > \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_{\theta'}, v})$  for any  $v$  and  $\mathcal{F}_L^G$ .

**Proof 3** As  $\theta \prec_\varphi \theta'$ , we have  $\varphi_\theta \Rightarrow \varphi_{\theta'}$  holds true and  $\varphi_{\theta'} \Rightarrow \varphi_\theta$  does not hold true at any node  $v$ . Thus, for any node  $v$  and any graph-temporal trajectory  $g \in \mathcal{B}_L^G$ , if  $(g, v) \models \varphi_\theta$ , then  $(g, v) \models \varphi_{\theta'}$ . Besides, for any node  $v$  there exists at least one graph-temporal trajectory  $\hat{g} \in \mathcal{B}_L^G$  such that  $(\hat{g}, v) \models \varphi_{\theta'}$  and  $(\hat{g}, v) \not\models \varphi_\theta$ . From Assumption 1,  $\hat{g}$  occurs with non-zero probability based on  $\mathcal{F}_L^G$ . Therefore,  $\mathbb{P}_{\mathcal{F}_L^G, \varphi_\theta, v} < \mathbb{P}_{\mathcal{F}_L^G, \varphi_{\theta'}, v}$ . Then according to Proposition 1,  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) > \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_{\theta'}, v})$  for any  $v$  and  $\mathcal{F}_L^G$ .

For a pGTL formula  $\varphi$ , we denote by  $\Theta_{sat}$  the set of parameter valuations  $\theta$  such that  $\varphi_\theta$  satisfies the coverage constraint. We further denote  $\Theta_{unsat} := \Theta \setminus \Theta_{sat}$ .

**Definition 13** A parameter valuation  $\theta$  in  $\Theta_{sat}$  is said to be minimal if there does not exist a parameter valuation  $\theta' \in \Theta_{sat}$  such that  $\theta' \prec_\varphi \theta$ . We define the minimal satisfying set  $\Theta_s$  as the set of minimal parameter valuations in  $\Theta_{sat}$ .

**Proposition 2** For a prior distribution  $\mathcal{F}_L^G$  and  $\Theta_s$ , if  $\theta^* = \arg \max_{\theta \in \Theta_{sat}} \frac{1}{|V|} \sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v})$ , then  $\theta^* \in \Theta_s$ .

**Proof 4** We prove Proposition 2 by contradiction. If  $\theta^* \notin \Theta_s$ , then from Definition 13 there exists  $\theta \in \Theta_{sat}$  such that  $\theta \neq \theta^*$  and  $\theta \prec_\varphi \theta^*$ . Then from Lemma 2 we have  $\mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) > \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_{\theta^*}, v})$  for any node  $v$ . Thus, we have  $\sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) > \sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_{\theta^*}, v})$ . But as  $\theta^* = \arg \max_{\theta \in \Theta_{sat}} \frac{1}{|V|} \sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v})$ , we have  $\sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_\theta, v}) \leq \sum_{v \in V} \mathcal{I}(\mathcal{F}_L^G, \bar{\mathcal{F}}_L^{\varphi_{\theta^*}, v})$ . Contradiction.

From Proposition 2, it can be seen that the optimal parameter valuation belongs to the minimal satisfying set.

#### 4.4 Information-Guided Identification of GTL Formulas

In this subsection, we present the algorithm for the information-guided identification of GTL formulas.

For a pGTL formula  $\varphi$ , suppose that the parameter valuation  $\theta = [\theta_1, \dots, \theta_z]$  belongs to a set  $\Theta = [\theta_1^{\min}, \theta_1^{\max}] \times \dots \times [\theta_z^{\min}, \theta_z^{\max}]$ , where  $\theta_i^{\min} \leq \theta_i^{\max}$  ( $i = 1, \dots, z$ ). We define the mapping  $\Pi : \Theta \rightarrow [0, 1]^z$  as follows: for each  $\theta \in \Theta$ ,  $\Pi(\theta) = \omega = [\omega_1, \dots, \omega_z]$ , where for each  $i \in [1, z]$ , we have

$$\omega_i = \begin{cases} (\theta_i - \theta_i^{\min}) / (\theta_i^{\max} - \theta_i^{\min}) & \text{if } \varrho(\theta_i, \varphi) = +; \\ (\theta_i^{\max} - \theta_i) / (\theta_i^{\max} - \theta_i^{\min}) & \text{if } \varrho(\theta_i, \varphi) = -. \end{cases}$$

In this way, we transform the set of parameter valuations  $\Theta$  to the hypercube  $[0, 1]^z$ , and for each  $i$  the pGTL formula is easier to be satisfied if  $\omega_i$  is increased. Under this transformation, we use  $\Omega_s$  to denote the set of parameter valuations transformed from the parameter valuations in the minimal satisfying set  $\Theta_s$  (we also call  $\Omega_s$  the *transformed* minimal satisfying set).

**Definition 14** *The Hausdorff directed distance from a set  $S$  to a set  $S'$  is defined as*

$$\hat{d}_H(S, S') = \max_{s \in S} \min_{s' \in S'} \max_{1 \leq i \leq z} d(s'_i, s_i),$$

where  $s = [s_1, \dots, s_z] \in S$ , the directed distance  $d(s'_i, s_i) = s_i - s'_i$ , if  $s_i > s'_i$ ; and  $d(s'_i, s_i) = 0$ , if  $s_i \leq s'_i$  ( $i = 1, \dots, z$ ).

For a pGTL formula  $\varphi$ , suppose that after several queries we obtain the sets  $\hat{\Omega}_{unsat}$  and  $\hat{\Omega}_{sat}$  of parameter valuations, where for each  $\omega \in \hat{\Omega}_{sat}$ ,  $\varphi_{\Pi^{-1}(\omega)}$  satisfies the coverage constraint; and for each  $\omega \in \hat{\Omega}_{unsat}$ ,  $\varphi_{\Pi^{-1}(\omega)}$  violates the coverage constraint (see Fig. 4). We use  $\hat{\Omega}_s$  to denote the set of minimal parameter valuations in  $\hat{\Omega}_{sat}$ .

**Definition 15** ( $\epsilon$ -approximation [20]) *A set of parameter valuations  $\hat{\Omega}_s$  is an  $\epsilon$ -approximation of the set  $\Omega_s$  if  $\hat{d}_H(\Omega_s, \hat{\Omega}_s) \leq \epsilon$ , where  $\hat{d}_H(\Omega_s, \hat{\Omega}_s)$  is the Hausdorff directed distance from  $\Omega_s$  to  $\hat{\Omega}_s$ .*

**Proposition 3** [20] *A set  $\hat{\Omega}_s$  of parameter valuations is an  $\epsilon$ -approximation of the transformed minimal satisfying set  $\Omega_s$  if  $\hat{d}_H(knee(\hat{\Omega}_{unsat}), \hat{\Omega}_s) \leq \epsilon$ , where  $knee(\hat{\Omega}_{unsat})$  denotes the set of knee points of  $\hat{\Omega}_{unsat}$  (a point in the boundary of  $\hat{\Omega}_{unsat}$  is called a knee point if by subtracting a positive number from any of its coordinates we obtain a point in the interior of  $\hat{\Omega}_{unsat}$ ).*

The algorithm for the information-guided parametric identification of GTL formulas is shown in Algorithm 2. The identification is performed in two steps. In the first step (Line 1 to Line 6), we approximate the minimal satisfying set  $\Theta_s$  (transformed minimal satisfying set  $\Omega_s$ ) by iteratively querying the parameter space and updating the sets of parameter valuations that satisfy and violate the coverage constraint, respectively. In the second step (Line 7 to Line 10), we compute the parameter valuation that maximizes the average information gain with the parameter valuations chosen from the approximated minimal satisfying set.

We first initialize  $\hat{\Omega}_{sat}$  and  $\hat{\Omega}_s$  to be the vector of ones, and  $\hat{\Omega}_{unsat}$  to be the vector of zeros. We use  $r$  to denote the maximal directed distance from the set of knee points  $knee(\hat{\Omega}_{unsat})$  to the set  $\hat{\Omega}_s$  (Line 3), and  $\omega$  to denote the knee point with the maximal directed distance to the set  $\hat{\Omega}_s$  (Line 4). After each query, we select the next query as  $\omega + r/2$ , which is guaranteed to lie neither in  $\hat{\Omega}_{sat}$  nor in  $\hat{\Omega}_{unsat}$  (we choose  $\omega + r/2$  in the manner of binary search). Then we update  $\hat{\Omega}_{sat}$  if  $\varphi_{\Pi^{-1}(\omega+r/2)}$  satisfies the coverage constraint, and update  $\hat{\Omega}_{unsat}$  if it violates the coverage constraint [20] (Line 5). And the same procedure repeats until an  $\epsilon$ -approximation of  $\Omega_s$  is achieved. Finally, we identify the GTL formula with the parameter valuation in the approximated minimal satisfying set that provides the highest average information gain (Line 7 to Line 10).

## 5 Case Studies

We illustrate our approaches on three studies, with Case Study 1 on the classification problem, and Case Study 2 and 3 on the identification problem. The data used in Case Study 1 and 2 are from the SLS 3D printer at UT Austin, recorded with FLIR 6701 MWIR stationary Infrared camera.

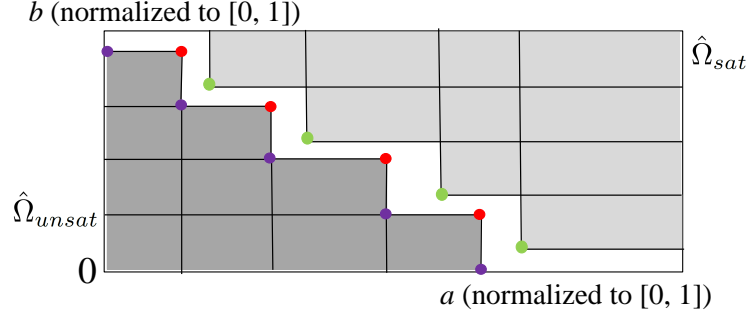


Figure 4: Querying the parameter space for a pGTL formula  $\exists^N \bigcirc_{y \leq a} (x \leq b)$  (with  $a$  and  $b$  dimensions shown). Green and red points correspond to parameter valuations of GTL formulas that satisfy and violate the coverage constraint respectively, while the purple points denote the knee points.

### 5.1 Case Study 1

The first case study is on classifying the graph patterns of ten tensile specimens built from SLS process (see Fig. 5). The tensile specimens have varying strengths, where the five stronger specimens labeled 1 have tensile strength above 46 MPa and the other five labeled -1 have tensile strength below 34 MPa.

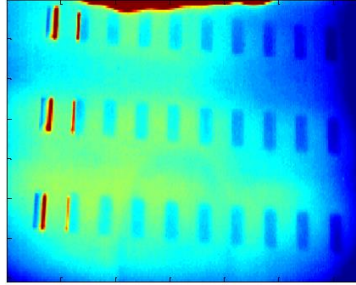


Figure 5: Infrared image of the rectangular cross-sections of the tensile specimens on the surface of the SLS powder bed [1].

We partition the fill region of each tensile specimen into 20 subregions ( $1200 \times 1200 \mu\text{m}^2$  for each subregion, with 210 layers), where each subregion is considered a node of a fully connected graph. The edge label  $y$  represents the Euclidean distance between the nodes (1 unit represents  $1200 \mu\text{m}$ ). As different layers are sintered at evenly spaced time instants, we use the layer indices to represent the time indices.

We use the following templates for type-I pGTL formulas (while other valid pGTL formulas can be also added to the set of templates, we choose the following ones as they are simple and sufficiently expressive for our applications):

$$\begin{aligned} \mathcal{P}^I = \{ & \square_{\geq i_1, \leq i_2} \exists^N \bigcirc_{\rho} \pi, & \diamond_{\geq i_1, \leq i_2} \exists^N \bigcirc_{\rho} \pi, \\ & \square_{\geq i_1, \leq i_2} \diamond_{\leq i_3} \exists^N \bigcirc_{\rho} \pi, & \diamond_{\geq i_1, \leq i_2} \square_{\leq i_3} \exists^N \bigcirc_{\rho} \pi, \\ & \square(\pi_1 \Rightarrow \square_{\leq i} \exists^N \bigcirc_{\rho} \pi_2), & \square(\pi_1 \Rightarrow \diamond_{\leq i} \exists^N \bigcirc_{\rho} \pi_2) \}, \end{aligned}$$

---

**Algorithm 2:** Information-guided parameter identification of GTL formulas.

---

```

1 Initialize  $\hat{\Omega}_{sat}, \hat{\Omega}_s, \hat{\Omega}_{unsat}, \hat{\mathcal{I}} \leftarrow 0$ 
2 while  $\hat{d}_H(knee(\hat{\Omega}_{unsat}), \hat{\Omega}_s) > \epsilon$  do
3    $r \leftarrow \max_{\omega \in knee(\hat{\Omega}_{unsat})} d(\omega, \hat{\Omega}_s)$ 
4    $\omega \leftarrow \arg \max_{\omega \in knee(\hat{\Omega}_{unsat})} d(\omega, \hat{\Omega}_s), \omega \leftarrow \omega + r/2$ 
5   Add  $\omega$  to  $\hat{\Omega}_{sat}$  if  $\varphi_{\Pi^{-1}(\omega)}$  satisfies the coverage constraint and add  $\omega$  to  $\hat{\Omega}_{unsat}$ 
   otherwise
6   Update  $\hat{\Omega}_s$  as the minimal satisfying set of  $\hat{\Omega}_{sat}$ 
7 for  $\omega \in \hat{\Omega}_s$  do
8    $\mathcal{I} \leftarrow ComputeIG(\mathcal{S}_L^G, \varphi_{\Pi^{-1}(\omega)}, \mathcal{F}_L^G)$ 
9   if  $\mathcal{I} > \hat{\mathcal{I}}$  then
10     $\hat{\mathcal{I}} \leftarrow \mathcal{I}, \hat{\omega} \leftarrow \omega$ 
11 return  $\varphi_{\Pi^{-1}(\hat{\omega})}$ 

```

---

where  $\pi, \pi_1$  and  $\pi_2$  are atomic node propositions in the form of  $x \geq c_1$  or  $x \leq c_1$  ( $c_1 \in \mathbb{R}$ ),  $\rho$  is an edge proposition in the form of  $y \leq c_2$  ( $c_2$  is a positive integer),  $N$  is a positive integer,  $i_1, i_2, i_3, i \in \mathbb{T}$  and  $i_1 < i_2$ .

We use the following templates for type-II pGTL formulas:

$$\mathcal{P}^{\text{II}} = \{ \exists^N \bigcirc_{\rho} \square_{\geq i_1, \leq i_2} \pi, \quad \exists^N \bigcirc_{\rho} \diamond_{\geq i_1, \leq i_2} \pi, \\ \exists^N \bigcirc_{\rho} \square_{\geq i_1, \leq i_2} \diamond_{\leq i_3} \pi, \quad \exists^N \bigcirc_{\rho} \diamond_{\geq i_1, \leq i_2} \square_{\leq i_3} \pi \},$$

where  $\pi, \rho, N, i_1, i_2$  and  $i_3$  are as described in  $\mathcal{P}^{\text{I}}$ .

We set  $m_{\text{th}} = 0.02$ ,  $\eta_{\text{th}} = 3$  and  $\hat{m}_{\text{th}} = 0.1$ . Using the approach illustrated in Sec. 3, we obtain the following GTL formula from  $\mathcal{P}^{\text{I}}$  and  $\mathcal{P}^{\text{II}}$  with zero nodal misclassification rate:

$$\varphi_{\theta^*, 1}^* = \square(T \geq 181.1 \Rightarrow \square_{\leq 6} \exists^8 \bigcirc_{y \leq 2} T \leq 198.0) \wedge \\ \square(T \leq 204.0 \Rightarrow \square_{\leq 2} \exists^3 \bigcirc_{y \leq 1} T \geq 181.8),$$

which means “(for any node) whenever the temperature is at least 181.1 C°, then for the next 6 layers there are always at least 8 nodes within distance of 2 where the temperature is at most 198.0 C°; whenever it is at most 204.0 C°, then for the next 2 layers there are always at least 3 nodes within distance of 1 where the temperature is at least 181.8 C°”.

$\varphi_{\theta^*, 1}^*$  is validated with another set of ten tensile specimens (five labeled 1 with tensile strength above 43 MPa and five labeled -1 with tensile strength below 37 MPa), with nodal misclassification rate of 8.33%.

## 5.2 Case Study 2

The second case study is on identifying informative patterns from data of SLS cooldown process (see Fig. 1 (a) in Sec. I). We record 16 graph temporal trajectories from  $7 \times 7$  grids of the powder bed, where each of the 49 cells ( $400 \times 400 \mu\text{m}^2$  for each cell) is considered a

node of a fully connected graph. The edge label  $y$  represents the Euclidean distance between the nodes (1 unit represents 400  $\mu\text{m}$ ).

We set  $p_{\text{th}} = 0.98$  and  $\epsilon = 0.05$ . Through Algorithm 2, we obtain the best type-I and type-II GTL formulas from  $\mathcal{P}^{\text{I}}$  and  $\mathcal{P}^{\text{II}}$  as (with the average information gain of 0.1563 and 0.0013, respectively, both with coverage rate of 100%):

$$\begin{aligned}\varphi_{\theta^*,2}^{\text{I}*} &= \Box(T \geq 183.4 \Rightarrow \Box_{\leq 3} \exists^2 \bigcirc_{y \leq 1} T \leq 182.8), \\ \varphi_{\theta^*,2}^{\text{II}*} &= \exists^4 \bigcirc_{y \leq 2} \Box_{\geq 3, \leq 8}(T \geq 178.7),\end{aligned}$$

where  $\varphi_{\theta^*,2}^{\text{I}*}$  means “(for any node) whenever the temperature is at least 183.4 C°, then for the next 3 time steps there are always at least 2 nodes within distance of 1 where the temperature is at most 182.8 C°”, and  $\varphi_{\theta^*,2}^{\text{II}*}$  reads as “(for any node) there exists at least 4 nodes within distance of 2 where the temperature is always at least 178.7 C° from time step 3 to 8” (each time step lasts for 33 milliseconds).

$\varphi_{\theta^*,2}^{\text{I}*}$  and  $\varphi_{\theta^*,2}^{\text{II}*}$  are validated with another set of 16 graph temporal trajectories recorded from another layer of the powder bed, both with coverage rate of 100%.

### 5.3 Case Study 3

The third case study is on identifying informative patterns from simulation data of a swarm of robots. We partition the workspace into 9 subregions (as shown in Fig. 6), where each subregion is considered a node of a fully connected graph. The edge label  $y$  represents the Euclidean distance between the centroids of the subregions. The probabilistic densities of the robots in the subregions are governed by a time-varying Markov chain [21].

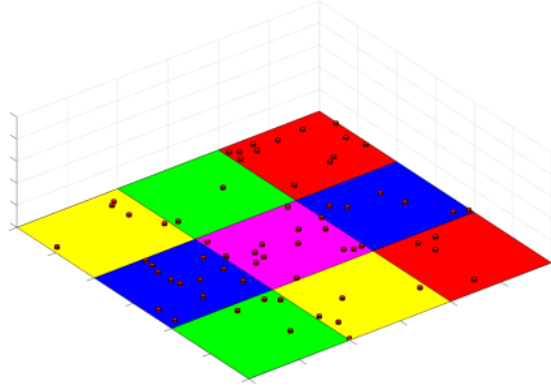


Figure 6: The swarm of 72 robots in the 9 sub-regions.

We randomly generate graph-temporal trajectories and randomly choose 10 from them that satisfy the following constraint: whenever the probabilistic density of a subregion reaches above 1/8, then for the next 2 time units there always exists at least one neighbor subregion within distance of 1 with probabilistic density below 1/9. Then we infer a GTL formula from the 10 graph-temporal trajectories using Algorithm 2.

We set  $p_{\text{th}} = 0.98$  and  $\epsilon = 0.05$ . Through Algorithm 2, we obtain the best type-I and type-II GTL formulas from  $\mathcal{P}^{\text{I}}$  and  $\mathcal{P}^{\text{II}}$  as (with the average information gain of 0.1 and



0.0043, respectively, both with coverage rate of 100%):

$$\begin{aligned}\varphi_{\theta^*,3}^{I*} &= \Box(x \geq 0.1894 \Rightarrow \Box_{\leq 2} \exists^1 \bigcirc_{y \leq 1} x \leq 0.1137), \\ \varphi_{\theta^*,3}^{II*} &= \exists^2 \bigcirc_{y \leq 1} \Box_{\geq 4, \leq 6}(x \geq 0.0379),\end{aligned}$$

where  $x$  is the probabilistic density in a subregion. It can be seen that  $\varphi_{\theta^*,3}^{I*}$  is different but similar with the set constraint.

$\varphi_{\theta^*,3}^{I*}$  and  $\varphi_{\theta^*,3}^{II*}$  are validated with another set of 10 randomly generated graph temporal trajectories that satisfy the set constraint, both with coverage rate of 100%.

## 6 Conclusion

We have introduced GTL and proposed the framework and algorithms to infer GTL formulas from data for classification and identification. For future work, we will consider more efficient methods for inferring more general forms of GTL formulas. In various network systems, specifications can be expressed in GTL, hence verification and controller synthesis can be also conducted with GTL specifications.

## References

- [1] S. Taylor, “Thermal history correlation with mechanical properties for polymer selective laser sintering (SLS),” Master Thesis, The University of Texas at Austin, Austin, US, 2017.
- [2] 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.
- [3] E. Asarin, A. Donzé, O. Maler, and D. Nickovic, “Parametric identification of temporal properties,” in *Proc. Second Int. Conf. Runtime Verification*, Berlin, Heidelberg, 2012, pp. 147–160.
- [4] X. Jin, A. Donze, J. V. Deshmukh, and S. A. Seshia, “Mining requirements from closed-loop control models,” in *Proc. Int. Conf. Hybrid Syst.: Comput. and Control*, 2013, pp. 43–52.
- [5] G. Bombara, C.-I. Vasile, F. Penedo, H. Yasuoka, and C. Belta, “A decision tree approach to data classification using signal temporal logic,” in *Proc. Int. Conf. Hybrid Syst.: Comput. and Control*. New York, NY, USA: ACM, 2016, pp. 1–10.
- [6] Z. Xu and A. A. Julius, “Census signal temporal logic inference for multiagent group behavior analysis,” *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 264–277, Jan. 2018.
- [7] B. Hoxha, A. Dokhanchi, and G. Fainekos, “Mining parametric temporal logic properties in model-based design for cyber-physical systems,” *International Journal on Software Tools for Technology Transfer*, 20.1 (2018): 79-93.
- [8] D. Neider and I. Gavran, “Learning linear temporal properties,” *Formal Methods in Computer Aided Design (FMCAD)*, pp. 1–10, 2018.

- [9] M. Vazquez-Chanlatte, S. Jha, A. Tiwari, M. K. Ho, and S. A. Seshia, “Learning task specifications from demonstrations,” in *NeurIPS*, 2018.
- [10] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, “The robotarium: A remotely accessible swarm robotics research testbed,” in *IEEE Inter. Conf. Robotics Automation (ICRA)*, May 2017, pp. 1699–1706.
- [11] I. Haghighi, A. Jones, Z. Kong, E. Bartocci, R. Gros, and C. Belta, “Spatel: A novel spatial-temporal logic and its applications to networked systems,” in *HSCC’15*. New York, NY, USA: ACM, 2015, pp. 189–198.
- [12] L. Bortolussi and L. Nenzi, “Specifying and monitoring properties of stochastic spatio-temporal systems in signal temporal logic,” in *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS ’14. ICST, Brussels, Belgium, Belgium: ICST, 2014, pp. 66–73.
- [13] B. Courcelle, “On the expression of graph properties in some fragments of monadic second-order logic,” in *Descriptive Complexity and Finite Models: Proceedings of a DIAMCS Workshop, chapter 2*. American Mathematical Society, 1997, pp. 33–57.
- [14] L. Cardelli, P. Gardner, and G. Ghelli, “A spatial logic for querying graphs,” in *Automata, Languages and Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 597–610.
- [15] Z. Xu, M. Ornik, A. A. Julius, and U. Topcu, “Information-guided temporal logic inference with prior knowledge,” in *Proc. Amer. Control Conf. (ACC)*, 2019. [Online]. Available: <https://arxiv.org/abs/1811.08846>
- [16] S. Chakraborty and J.-P. Katoen, “Parametric LTL on Markov chains,” in *Theoretical Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 207–221.
- [17] O. Kupferman and M. Y. Vardi, “Model checking of safety properties,” *Form. Methods Syst. Des.*, vol. 19, no. 3, pp. 291–314, Oct. 2001.
- [18] Eberhart and Y. Shi, “Particle swarm optimization: developments, applications and resources,” in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, May 2001, pp. 81–86 vol. 1.
- [19] T. Nghiem, S. Sankaranarayanan, G. Fainekos, F. Ivancić, A. Gupta, and G. J. Pappas, “Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems,” in *Proc. HSCC’10*. ACM, 2010, pp. 211–220.
- [20] J. Legriël, C. Le Guernic, S. Cotton, and O. Maler, “Approximating the pareto front of multi-criteria optimization problems,” in *Tools and Algorithms for the Construction and Analysis of Systems*, J. Esparza and R. Majumdar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 69–83.
- [21] N. Demirer, M. E. Chamie, and B. Akmeë, “Safe markov chains for on/off density control with observed transitions,” *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1442–1449, May 2018.