

Information-Guided Temporal Logic Inference with Prior Knowledge

Zhe Xu, Melkior Ornik, A. Agung Julius, Ufuk Topcu *

November 22, 2018

Abstract

This paper investigates the problem of inferring knowledge from data so that the inferred knowledge is interpretable and informative to humans who have prior knowledge. Given a dataset as a collection of system trajectories, we infer parametric linear temporal logic (pLTL) formulas that are informative and satisfied by the trajectories in the dataset with high probability. The informativeness of the inferred formula is measured by the information gain with respect to given prior knowledge represented by a prior probability distribution. We first present two algorithms to compute the information gain with a focus on two types of prior probability distributions: stationary probability distributions and probability distributions expressed by discrete time Markov chains. Then we provide a method to solve the inference problem for a subset of pLTL formulas with polynomial time complexity with respect to the number of Boolean connectives in the formula. We provide implementations of the proposed approach on explaining anomalous patterns, patterns changes and explaining the policies of Markov decision processes.

1 Introduction

Inferring human-interpretable knowledge from execution trajectories of a system is important in many applications. Such knowledge, for example, may represent behaviors that deviate from the expected behaviors according to some prior knowledge. These deviations may unveil some unknown patterns of the system, or indicate that changes or anomalies have

*Zhe Xu and Melkior Ornik are with the Institute for Computational Engineering and Sciences (ICES), University of Texas, 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: zhxu@utexas.edu, mornik@ices.utexas.edu, juliua2@rpi.edu, utopcu@utexas.edu.

occurred in the system. As a simple example, suppose that we are given a recently collected dataset of weather conditions on **consecutive** days in a certain geographical region, as shown in Fig. 1. Furthermore, suppose that we also have prior knowledge on historical weather conditions in the region, represented as a discrete time Markov chain as shown in Fig. 2. We raise the following question: What knowledge can we infer from the dataset so that it is *interpretable* and *informative* to humans who have the possibly outdated impression (i.e. prior knowledge)?

The interpretability of the inferred knowledge represents the extent that humans can understand the knowledge. In the past decade, there has been a growing interest in inferring temporal logic formulas from system trajectories [1, 2, 3, 4]. The temporal logic formulas can express features and patterns in a form that resembles natural languages [5] and thus improve the interpretability of the inferred knowledge.

The informativeness represents the extent to which the inferred knowledge deviates from prior knowledge. In the example on weather conditions, suppose that we are given two candidate temporal logic formulas: one reads as “whenever it is rainy, the next day will be sunny” and the other one reads as “whenever it is sunny, the next day will be rainy”. While both formulas are consistent with the data as shown in Fig. 1, clearly the second one is more informative in describing how the weather patterns have deviated from the prior knowledge as shown in Fig. 2.



Figure 1: The dataset of weather conditions on consecutive days in a certain geographical region.

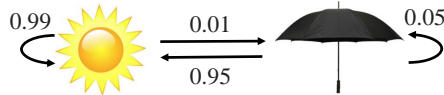


Figure 2: The prior knowledge represented as a discrete time Markov chain (with two simplified weather conditions: sunny and rainy).

There are various methods for inferring temporal logic formulas from data. For example, the authors in [5] developed an inference method to automatically select both the structure and the parameters of signal temporal logic (STL) formulas that classify the trajectories between a desired set and an undesired set. In [6], the authors proposed a decision-tree approach to infer STL formulas for classification. In [7], the authors presented methods to infer linear temporal logic formulas from data through a reduction to a series of satisfiability problems in propositional Boolean logic. However, to the best of our knowledge, none of the existing approaches performs the inference task while considering the informativeness

of the inferred formulas over prior knowledge.

In this paper, we propose a theoretical framework and algorithms for the information-guided temporal logic inference. We infer a parametric linear temporal logic formula that is consistent with a set of trajectories and provides a high information gain (the notion of information gain will be formalized in Section 3) over a given prior probability distribution. We conduct three case studies as implementations of the proposed approach in the following applications.

- *Explaining anomalous patterns and pattern changes:* If we assume that normal behavior occur with high probability based on the prior probability distribution, then the inferred formulas can be used to explain anomalous patterns “hidden” in the dataset. If the prior probability distribution represents outdated knowledge or impressions, then the inferred formulas can be used to explain pattern changes at present.
- *Explaining policies of Markov decision processes:* If the dataset consists of observed trajectories of a Markov decision process (MDP), then the inferred temporal logic formulas may be used as explanations of the policies of the MDP.

2 Parametric Linear Temporal Logic

In this section, we present an overview of parametric linear temporal logic (pLTL) [8, 9]. We start with the syntax and semantics of pLTL. The domain $\mathbb{B} = \{\top, \perp\}$ (\top and \perp represents True and False respectively) is the Boolean domain and the time index set is a discrete set of natural numbers $\mathbb{T} = \{1, 2, \dots\}$. We assume that there is an underlying system \mathcal{H} . The state s of the system \mathcal{H} belongs to a finite set of states S . A trajectory $s_{1:L} = s_1 s_2 \dots s_L$ describing an evolution of the system \mathcal{H} is a function from \mathbb{T} to S . A set $\mathcal{AP} = \{\pi_1, \pi_2, \dots, \pi_n\}$ is a set of atomic predicates. $\mathcal{L} : S \rightarrow 2^{\mathcal{AP}}$ is a labeling function assigning a subset of atomic predicates in \mathcal{AP} to each state $s \in S$.

The syntax of the pLTL formula is defined recursively as follows:

$$\phi := \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \bigcirc\phi \mid \phi_1 \mathcal{U}\phi_2 \mid \Diamond_{\sim i}\phi,$$

where π is an atomic predicate, \neg and \wedge stand for negation and conjunction respectively, \bigcirc and \mathcal{U} are temporal operators representing “next” and “until” respectively, $\Diamond_{\sim i}$ ($i \in \mathbb{T}$) is a parametrized temporal operator representing “parametrized eventually”, where $\sim \in \{\geq, \leq\}$, i is a temporal parameter. We can also derive \vee (disjunction), \Diamond (eventually), \Box (always), \mathcal{R} (release), $\Box_{\sim i}$ (parametrized always), $\mathcal{U}_{\sim i}$ (parametrized until), $\mathcal{R}_{\sim i}$ (parametrized release) from the above-mentioned operators as described in [8]. We can also derive $\Diamond_{\geq i_1, \leq i_2}$ and

$\Box_{\geq i_1, \leq i_2} (i_1 < i_2)$ as

$$\begin{aligned}\Diamond_{\geq i_1, \leq i_2} \phi &= \Diamond_{\geq i_1} \phi \wedge \Diamond_{\leq i_2} \phi, \\ \Box_{\geq i_1, \leq i_2} \phi &= \Box_{\geq i_1} \phi \wedge \Box_{\leq i_2} \phi.\end{aligned}$$

Next, we introduce the Boolean semantics of a pLTL formula in the strong and the weak view [10], which is used in evaluating the satisfaction or violation of pLTL formula by trajectories of finite length.

The satisfaction relation $(s_{1:L}, k) \models_S \phi$ for trajectory $s_{1:L}$ at time index k with respect to a pLTL formula ϕ as Boolean semantics in the strong view is defined recursively as follows:

$$\begin{aligned}(s_{1:L}, k) \models_S \pi &\text{ iff } k \leq L \text{ and } \pi \in \mathcal{L}(s_k), \\ (s_{1:L}, k) \models_S \neg \phi &\text{ iff } (s_{1:L}, k) \not\models_W \phi, \\ (s_{1:L}, k) \models_S \phi_1 \wedge \phi_2 &\text{ iff } (s_{1:L}, k) \models_S \phi_1 \\ &\text{ and } (s_{1:L}, k) \models_S \phi_2 \\ (s_{1:L}, k) \models_S \bigcirc \phi &\text{ iff } (s_{1:L}, k+1) \models_S \phi, \\ (s_{1:L}, k) \models_S \phi_1 \mathcal{U} \phi_2 &\text{ iff } \exists k' \geq k, \text{ s.t. } (s_{1:L}, k') \models_S \phi_2, \\ &\quad (s_{1:L}, k'') \models_S \phi_1, \forall k'' \in [k, k'] \\ (s_{1:L}, k) \models_S \Diamond_{\sim i} \phi &\text{ iff } \exists k' \sim k+i, \text{ s.t. } (s_{1:L}, k') \models_S \phi.\end{aligned}$$

The satisfaction relation $(s_{1:L}, k) \models_W \phi$ as Boolean semantics in the weak view is defined recursively as follows:

$$\begin{aligned}(s_{1:L}, k) \models_W \pi &\text{ iff either of the following holds :} \\ &\quad 1) k > L; \ 2) k \leq L \text{ and } \pi \in \mathcal{L}(s_k), \\ (s_{1:L}, k) \models_W \neg \phi &\text{ iff } (s_{1:L}, k) \not\models_S \phi,\end{aligned}$$

while the Boolean semantics of the other logical and temporal operators can be obtained from those in the strong view by replacing each \models_S with \models_W .

If the satisfaction relations are evaluated at time index $k = 1$, then we write $s_{1:L} \models_S \phi$ or $s_{1:L} \models_W \phi$ for brevity.

Intuitively, if a trajectory of finite length can be extended to infinite length, then the strong view indicates that the truth value of the formula on the infinite-length trajectory is already “determined” on the trajectory of finite length, while the weak view indicates that it may not be “determined” yet [10]. As an example, a trajectory $s_{1:3} = s_1 s_2 s_3$ is not possible to strongly satisfy $\phi = \Box_{\leq 5} \pi$, but $s_{1:3}$ is possible to strongly violate ϕ , i.e., $(s_{1:L}, k) \models_S \neg \phi$ is possible.

Trajectories of finite length are sufficient to strongly satisfy (resp. violate) *syntactically*

co-safe (resp. safe) pLTL formulas [11], which are defined in the following definitions.

Definition 1 *The syntax of the syntactically co-safe pLTL formula is defined recursively as follows:*

$$\begin{aligned} \phi := & \top \mid \pi \mid \neg\pi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc\phi \mid \Diamond\phi \mid \phi_1 \mathcal{U}\phi_2 \\ & \mid \Diamond_{\sim i}\phi \mid \Box_{\leq i}\phi \mid \phi_1 \mathcal{U}_{\sim i}\phi_2 \mid \phi_1 \mathcal{R}_{\leq i}\phi_2. \end{aligned}$$

Definition 2 *The syntax of the syntactically safe pLTL formula is defined as follows:*

$$\begin{aligned} \phi := & \perp \mid \pi \mid \neg\pi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc\phi \mid \Box\phi \mid \phi_1 \mathcal{R}\phi_2 \\ & \mid \Diamond_{\leq i}\phi \mid \Box_{\sim i}\phi \mid \phi_1 \mathcal{U}_{\leq i}\phi_2 \mid \phi_1 \mathcal{R}_{\sim i}\phi_2. \end{aligned}$$

There are pLTL formulas that are neither syntactically co-safe nor syntactically safe pLTL formulas, such as $\Box\Diamond_{\geq i}\pi$. These pLTL formulas can neither be strongly satisfied nor strongly violated by trajectories of finite length. As we intend to infer pLTL formulas from trajectories of finite length (as in a dataset), in this paper we only focus on syntactically co-safe or syntactically safe pLTL formulas, which will be referred to as (co-)safe pLTL formulas for brevity.

In the following, for simplicity we require that the trajectory of finite length is sufficiently long so that it can strongly satisfy (resp. violate) the corresponding syntactically co-safe (resp. safe) formula. With slight abuse of notation, we use the notations $s_{1:L} \models \phi$ and $s_{1:L} \not\models \phi$ to denote $s_{1:L} \models_S \phi$ and $s_{1:L} \not\models_S \phi$ if ϕ is syntactically co-safe, and to denote $s_{1:L} \models_W \phi$ and $s_{1:L} \not\models_W \phi$ if ϕ is syntactically safe. Additionally, the word “satisfy” without the modifying adverb “strongly” or “weakly” is meant in the strong (resp. weak) view if the corresponding formula ϕ is syntactically co-safe (resp. safe). If a pLTL formula ϕ is both syntactically co-safe and syntactically safe, then the strong view and the weak view are equivalent for ϕ as we require that the trajectory of finite length is sufficiently long so that it can strongly satisfy and strongly violate the formula.

The size of a (co-)safe pLTL formula ϕ , denoted as $\ell(\phi)$, is defined as the number of Boolean connectives in ϕ . Note that logically equivalent formulas may have different sizes.

3 Problem Formulation

We denote \mathcal{B}_L as the set of all possible trajectories with length L of the underlying system \mathcal{H} . We are also given a dataset $\mathcal{S}_L = \{\hat{s}_{1:L}^1, \dots, \hat{s}_{1:L}^m\} \subset \mathcal{B}_L$ as a collection of trajectories and a prior probability distribution over \mathcal{B}_L .

Notation 1 *We use $\mathcal{F}_L : \mathcal{B}_L \rightarrow [0, 1]$ to denote the prior probability distribution, $\mathbb{P}_{\mathcal{B}_L, \phi}$ to denote the probability of a trajectory $s_{1:L}$ satisfying ϕ in the set \mathcal{B}_L based on \mathcal{F}_L , and*

$\bar{\mathbb{P}}_{\mathcal{S}_L, \phi} := |\{\hat{s}_{1:L} : \hat{s}_{1:L} \in \mathcal{S}_L, \hat{s}_{1:L} \models \phi\}|/|\mathcal{S}_L|$ to denote the **empirical** probability of a trajectory $\hat{s}_{1:L}$ satisfying ϕ in the dataset \mathcal{S}_L (i.e. the ratio of the number of trajectories in \mathcal{S}_L that satisfy ϕ), where $|S|$ denotes the cardinality of a set S .

Assumption 1 We assume that every trajectory in \mathcal{B}_L occurs with non-zero probability according to \mathcal{F}_L .

Definition 3 Given a prior probability distribution \mathcal{F}_L that satisfies Assumption 1, we define $\bar{\mathcal{F}}_L^\phi : \mathcal{B}_L \rightarrow [0, 1]$ as the estimated posterior probability distribution given the pLTL formula ϕ and the dataset \mathcal{S}_L , which is expressed as

$$\bar{\mathcal{F}}_L^\phi(s_{1:L}) = \begin{cases} \frac{\mathcal{F}_L(s_{1:L})\bar{\mathbb{P}}_{\mathcal{S}_L, \phi}}{\bar{\mathbb{P}}_{\mathcal{B}_L, \phi}} & \text{if } s_{1:L} \models \phi, \\ \frac{\mathcal{F}_L(s_{1:L})\bar{\mathbb{P}}_{\mathcal{S}_L, \neg\phi}}{\bar{\mathbb{P}}_{\mathcal{B}_L, \neg\phi}} & \text{if } s_{1:L} \not\models \phi. \end{cases}$$

Remark 1 According to Assumption 1, for any $s_{1:L} \in \mathcal{B}_L$, if $s_{1:L} \models \phi$, then $\bar{\mathbb{P}}_{\mathcal{B}_L, \phi} > 0$; if $s_{1:L} \not\models \phi$, then $\bar{\mathbb{P}}_{\mathcal{B}_L, \neg\phi} > 0$.

Remark 2 We require that

$$\sum_{s_{1:L} \in \mathcal{B}_L, s_{1:L} \models \phi} \bar{\mathcal{F}}_L^\phi(s_{1:L}) = \bar{\mathbb{P}}_{\mathcal{S}_L, \phi},$$

which means the formula ϕ is true with the same empirical probability in \mathcal{S}_L as the probability in \mathcal{B}_L based on $\bar{\mathcal{F}}_L^\phi$. Then the expression of $\bar{\mathcal{F}}_L^\phi$ can be derived using Bayes' theorem.

Remark 3 If $\bar{\mathbb{P}}_{\mathcal{B}_L, \phi} = \bar{\mathbb{P}}_{\mathcal{S}_L, \phi}$, then $\bar{\mathcal{F}}_L^\phi(s_{1:L}) = \mathcal{F}_L(s_{1:L})$ for any $s_{1:L} \in \mathcal{B}_L$, which means that the estimated posterior probability distribution given ϕ and the dataset \mathcal{S}_L is the same as the prior probability distribution if ϕ is true with the same empirical probability in \mathcal{S}_L as the probability in \mathcal{B}_L based on \mathcal{F}_L . Especially, if $\phi = \top$ (resp. $\phi = \perp$), then $\bar{\mathbb{P}}_{\mathcal{B}_L, \phi} = \bar{\mathbb{P}}_{\mathcal{S}_L, \phi} = 1$ (resp. $\bar{\mathbb{P}}_{\mathcal{B}_L, \neg\phi} = \bar{\mathbb{P}}_{\mathcal{S}_L, \neg\phi} = 1$), so $\bar{\mathcal{F}}_L^\phi(s_{1:L}) = \mathcal{F}_L(s_{1:L})$ for any $s_{1:L} \in \mathcal{B}_L$, which means that the estimated posterior probability distribution given \top (\perp) and any dataset \mathcal{S}_L is the same as the prior probability distribution.

Definition 4 We define

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

as the **information gain** when the prior probability distribution \mathcal{F}_L is updated to the estimated posterior probability distribution $\bar{\mathcal{F}}_L^\phi$, where

$$D_{\text{KL}}(\bar{\mathcal{F}}_L^\phi || \mathcal{F}_L) = \sum_{s_{1:L} \in \mathcal{B}_L} \bar{\mathcal{F}}_L^\phi(s_{1:L}) \log \frac{\bar{\mathcal{F}}_L^\phi(s_{1:L})}{\mathcal{F}_L(s_{1:L})}$$

is the Kullback-Leibler divergence from \mathcal{F}_L to $\bar{\mathcal{F}}_L^\phi$.

Problem 1 Given a dataset $\mathcal{S}_L = \{\hat{s}_{1:L}^1, \dots, \hat{s}_{1:L}^m\}$, a prior probability distribution \mathcal{F}_L , real constant $p_{\text{th}} \in (0, 1]$ and integer constant $\ell_{\text{th}} \in (0, \infty)$, we construct a (co-) safe pLTL formula ϕ that maximizes the information gain $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ while satisfying the following two constraints:

- coverage constraint: $\bar{\mathbb{P}}_{\mathcal{S}_L, \phi} \geq p_{\text{th}}$, i.e., the trajectories in \mathcal{S}_L should satisfy ϕ with empirical probability at least p_{th} ;
- size constraint: $\ell(\phi) \leq \ell_{\text{th}}$, i.e., the size of ϕ should not exceed ℓ_{th} .

Intuitively, the objective is to make the inferred pLTL formula informative over the prior probability distributions. The coverage constraint is to make the inferred pLTL formula ϕ consistent (with probability no less than p_{th}) with the dataset. The size constraint is to make the inferred pLTL formula **concise**, as an unnecessarily long and complicated formula is too specific and interpretability is compromised.

4 Computation of Information Gain with Prior Probability Distributions

To solve Problem 1, one needs to compute the information gain for a (co-) safe pLTL formula. The following proposition relates $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ to the probabilities $\mathbb{P}_{\mathcal{B}_L, \phi}$ and $\bar{\mathbb{P}}_{\mathcal{S}_L, \phi}$.

Proposition 1 The information gain $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ satisfies

$$\begin{aligned} \mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi) = & \frac{1}{L} (\bar{\mathbb{P}}_{\mathcal{S}_L, \phi} \log(\bar{\mathbb{P}}_{\mathcal{S}_L, \phi}) - \bar{\mathbb{P}}_{\mathcal{S}_L, \phi} \log(\mathbb{P}_{\mathcal{B}_L, \phi}) \\ & + (\bar{\mathbb{P}}_{\mathcal{S}_L, \neg\phi}) \log(\bar{\mathbb{P}}_{\mathcal{S}_L, \neg\phi}) - (\bar{\mathbb{P}}_{\mathcal{S}_L, \neg\phi}) \log(\mathbb{P}_{\mathcal{B}_L, \neg\phi})). \end{aligned}$$

proof See Appendix.

In the following two subsections, we present how to recursively compute the information gain through Proposition 1 for two specific types of prior probability distributions.

4.1 Computation of Information Gain for Stationary Prior Probability Distributions

In this subsection, we present an algorithm to compute the information gain for stationary prior probability distributions.

We define a stationary prior probability distribution $\mathcal{F}_L : \mathcal{B}_L \rightarrow [0, 1]$ as follows: for each trajectory $s_{1:L} \in \mathcal{B}_L$, the probability $\mathcal{F}_L(s_{1:L}) = \mathcal{F}(s_1)\mathcal{F}(s_2)\dots\mathcal{F}(s_L)$, where \mathcal{F} is the probability distribution on S which remains the same for any time index k .

Definition 5 A deterministic finite automaton (DFA) is a tuple $\mathcal{A} = (\mathcal{Q}, q_0, \Sigma, \delta, \text{Acc})$ where $\mathcal{Q} = \{q_0, q_1, \dots, q_K\}$ is a finite set of states, q_0 is the initial state, Σ is the alphabet, $\delta : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is the transition relation, $\text{Acc} \subseteq 2^{\mathcal{Q}}$ is a finite set of accepting states.

For any syntactically co-safe (resp. safe) pLTL formula ϕ , a DFA $\mathcal{A}^\phi = (\mathcal{Q}, q_0, 2^{\mathcal{A}^P}, \delta, \text{Acc}^\phi)$ (resp. $\mathcal{A}^{\neg\phi} = (\mathcal{Q}, q_0, 2^{\mathcal{A}^P}, \delta, \text{Acc}^{\neg\phi})$) can be constructed with input alphabet $2^{\mathcal{A}^P}$ that accepts all and only trajectories that strongly satisfy ϕ (resp. strongly violate ϕ) [11].

Algorithm 1 Information gain for stationary prior probability distributions.

```

1: Input:  $\mathcal{S}_L = \{\hat{s}_{1:L}^1, \dots, \hat{s}_{1:L}^m\}, \phi, \mathcal{F}_L$ 
2:  $\beta \leftarrow \mathbb{P}_{\mathcal{S}_L, \phi}$ 
3: Obtain the DFA  $\mathcal{A}^\phi$  (if  $\phi$  is syntactically co-safe) or  $\mathcal{A}^{\neg\phi}$  (if  $\phi$  is syntactically safe)
4: for  $k = 0$  to  $K$  do
5:   if  $q_k \in \text{Acc}^\phi$  (resp.  $\text{Acc}^{\neg\phi}$ ) then  $p(L, q_k) \leftarrow 1$ 
6:   else  $p(L, q_k) \leftarrow 0$ 
7:   end if
8: end for
9: for  $\ell = L$  to  $2$ ,  $j = 0$  to  $K$  do
10:   $p(\ell - 1, q_j) \leftarrow \sum_{k=0}^K c_{j,k} p(\ell, q_k)$ 
11: end for
12: if  $\phi$  is syntactically co-safe then  $\gamma \leftarrow p(1, q_0)$ 
13: else  $\gamma \leftarrow 1 - p(1, q_0)$ 
14: end if
15: return  $\mathcal{I} = \frac{1}{L}(\beta \log(\frac{\beta}{\gamma}) + (1 - \beta) \log(\frac{1-\beta}{1-\gamma}))$ 

```

Proposition 2 For a (co-)safe pLTL formula ϕ and a stationary prior probability distribution \mathcal{F}_L , Algorithm 1 returns $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$.

proof See Appendix.

Remark 4 The time complexity of Algorithm 1 is $O(LK^2)$, where $K + 1$ is the number of states of the DFA \mathcal{A}^ϕ . For different (co-)safe pLTL formulas with the same predicates, same temporal operators but different temporal parameters, the computational cost can be reduced by storing the results of computation for one formula and reusing them in the computation for another formula. For example, for $\phi = \Box \Diamond_{\leq i} \pi$, we have $(K = i + 1)$

$$\mathbf{p}_L^\phi(\ell - 1, \cdot) = \begin{bmatrix} \Pi_{i+1} & \mathbb{P}(\neg\pi) \mathbf{e}_{i+1}^{i+1T} \\ \mathbf{0}_{1 \times (i+1)} & 1 \end{bmatrix} \cdot \mathbf{p}_L^\phi(\ell, \cdot), \quad (1)$$

where $\mathbf{p}_L^\phi(\ell, \cdot) := [p_L^\phi(\ell, q_0), p_L^\phi(\ell, q_1), \dots, p_L^\phi(\ell, q_K)]^T$, $\mathbb{P}(\pi)$ denotes the stationary probability of a state satisfying the predicate π , \mathbf{e}_i^j denotes a row vector where the j th entry is one and all other entries are zeros, and

$$\Pi_{i+1} = \begin{bmatrix} \mathbb{P}(\pi) & \mathbb{P}(\neg\pi) & 0 & \dots & 0 \\ \mathbb{P}(\pi) & 0 & \mathbb{P}(\neg\pi) & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbb{P}(\pi) & 0 & 0 & \dots & \mathbb{P}(\neg\pi) \\ \mathbb{P}(\pi) & 0 & 0 & \dots & 0 \end{bmatrix}_{(i+1) \times (i+1)}. \quad (2)$$

Then we have

$$\mathbf{p}_L^\phi(1, \cdot) = \begin{bmatrix} \Pi_{i+1} & \mathbb{P}(\neg\pi)\mathbf{e}_{i+1}^{i+1T} \\ \mathbf{0}_{1 \times (i+1)} & 1 \end{bmatrix}^{L-1} \cdot \mathbf{p}_L^\phi(L, \cdot). \quad (3)$$

The computation burden mainly lies in computing Π_{i+1}^{L-1} .

For $\phi' = \square \Diamond_{\leq i+1} \pi$, the recursion becomes the following ($K = i + 2$):

$$\mathbf{p}_L^{\phi'}(1, \cdot) = \begin{bmatrix} \Pi_{i+2} & \mathbb{P}(\neg\pi)\mathbf{e}_{i+2}^{i+2T} \\ \mathbf{0}_{1 \times (i+2)} & 1 \end{bmatrix}^{L-1} \cdot \mathbf{p}_L^{\phi'}(L, \cdot), \quad (4)$$

where

$$\Pi_{i+2} = \begin{bmatrix} \Pi_{i+1} & \mathbb{P}(\neg\pi)\mathbf{e}_{i+1}^{i+1T} \\ \mathbb{P}(\pi)\mathbf{e}_{i+1}^1 & 0 \end{bmatrix}. \quad (5)$$

If we reuse the results of Π_{i+1}^{L-1} when computing Π_{i+2}^{L-1} , then the extra cost of computing (4) is only $O(LK)$.

4.2 Computation of Information Gain for Prior Probability Distributions Governed by DTMCs

In this subsection, we present an algorithm to compute the information gain for prior probability distributions governed by discrete-time Markov chains, which can be seen as generalizations of stationary prior probability distributions.

Definition 6 A discrete-time Markov chain (DTMC) is defined by a tuple $\mathcal{M} = (S, S_0, P, \mathcal{AP}, \mathcal{L})$, where $S = \{s^1, s^2, \dots, s^H\}$ is a finite set of states, $S_0 \subset S$ is the initial set of states, $P : S \times S \rightarrow [0, 1]$ is the transition probability, $P(s, s') \geq 0$ for all $s, s' \in S$ and $\sum_{s' \in S} P(s, s') = 1$, \mathcal{AP} is a set of atomic predicates, and $\mathcal{L} : S \rightarrow 2^{\mathcal{AP}}$ is a labeling function.

In the following, the DTMC that governs the prior probability distribution will be referred to as the *prior* DTMC.

Definition 7 (Product Automaton) Let $\mathcal{M} = (S, S_0, P, \mathcal{AP}, \mathcal{L})$ be a DTMC and $\mathcal{A}^\phi = (\mathcal{Q}, q_0, 2^{\mathcal{AP}}, \delta, \text{Acc}^\phi)$ be a DFA. The product automaton $\mathcal{M}_p^\phi := \mathcal{M} \otimes \mathcal{A}^\phi = (S_p, S_{0p}, P_p, \mathcal{L}_p, \text{Acc}_p^\phi)$ is a tuple such that

- $S_p = S \times \mathcal{Q}$ is a finite set of states;
- S_{0p} is the initial set of states where for each $s_{0p} = (s^i, q) \in S_{0p}$, $s^i \in S_0$, $q = \delta(q_0, \mathcal{L}(s^i))$;
- $P_p((s, q), (s', q')) = \begin{cases} P(s, s') & \text{if } q' = \delta(q, \mathcal{L}(s')); \\ 0 & \text{otherwise;} \end{cases}$
- $\mathcal{L}_p((s, q)) = \{q\}$ is a labeling function; and
- $\text{Acc}_p^\phi = S \times \text{Acc}^\phi$ is a finite set of accepting states.

Proposition 3 For a (co-)safe pLTL formula ϕ and a prior DTMC \mathcal{M} , Algorithm 2 returns $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$.

proof See Appendix.

Remark 5 The time complexity of Algorithm 2 is $O(LK^2H^2)$, where H is the number of states of \mathcal{M} . By storing the results of computation for one (co-)safe pLTL formula, the extra cost of computation for another (co-) safe pLTL formula with the same predicates, same temporal operators but different temporal parameters is $O(LKH)$.

5 Information-Guided pLTL Inference

In this section, we present the algorithm for our information-guided temporal logic inference approach. Computing the information gain for a (co-)safe pLTL formula is the essential step for the inference. For a (co-)safe pLTL formula that contains Boolean connectives (i.e., conjunctions and disjunctions), the computation of the information gain is expensive as the number of states of the DFA is exponential with respect to the size of the formula [12].

To reduce the computational cost, we adopt a heuristic method to solve Problem 1 for a subset of (co-)safe pLTL formulas. At first, we infer a set of (co-)safe pLTL formulas that do not contain Boolean connectives. Such pLTL formulas will be referred to as *primitive* pLTL formulas. Each of the inferred primitive pLTL formula, denoted as ϕ_j^1 , represents a *subpattern* that provides the maximal information gain (computed by the algorithms in

Algorithm 2 Information gain for prior probability distributions governed by DTMCs.

```

1: Input:  $\mathcal{S}_L = \{\hat{s}_{1:L}^1, \dots, \hat{s}_{1:L}^m\}, \phi, \mathcal{M}$ 
2:  $\beta \leftarrow \bar{\mathbb{P}}_{\mathcal{S}_L, \phi}$ 
3: Obtain the product automaton  $\mathcal{M}_p^\phi$  (if  $\phi$  is syntactically co-safe) or  $\mathcal{M}_p^{\neg\phi}$  (if  $\phi$  is syntactically safe)
4: for  $k = 0$  to  $K$  do
5:   if  $q_k \in \text{Acc}^\phi$  (resp.  $\text{Acc}^{\neg\phi}$ ) then
6:      $\forall i \in [1, H], p(L, s^i, q_k) \leftarrow 1$ 
7:   else  $\forall i \in [1, H], p(L, s^i, q_k) \leftarrow 0$ 
8:   end if
9: end for
10: for all  $j, k, i_1, i_2$  do
11:    $\mathbf{C}_{j,k}^p(i_1, i_2) \leftarrow P_p((s^{i_1}, q_j), (s^{i_2}, q_k))$ 
12: end for
13: for  $\ell = L$  to  $2, j = 0$  to  $K, i_1 = 1$  to  $H$  do
14:    $p(\ell - 1, s^{i_1}, q_j) \leftarrow \sum_{k=0}^K \sum_{i_2=1}^H \mathbf{C}_{j,k}^p(i_1, i_2) \cdot p(\ell, s^{i_2}, q_k)$ 
15: end for
16: if  $\phi$  is syntactically co-safe then
17:    $\gamma \leftarrow \sum_{s^i \in S_0} p_{\text{int}}(s^i) p(1, s^i, q_0)$ 
18: else  $\gamma \leftarrow \sum_{s^i \in S_0} p_{\text{int}}(s^i) (1 - p(1, s^i, q_0))$ 
19: end if
20: return  $\mathcal{I} = \frac{1}{L} (\beta \log(\frac{\beta}{\gamma}) + (1 - \beta) \log(\frac{1-\beta}{1-\gamma}))$ 

```

Section 4) with a certain temporal operator. We then form a *pattern* consisting of the subpatterns in the form of $\phi^1 = \phi_1^1 \wedge \phi_2^1 \wedge \dots \wedge \phi_q^1$. If ϕ^1 does not satisfy the coverage constraint (i.e., $\bar{\mathbb{P}}_{\mathcal{S}_L, \phi^1} < p_{\text{th}}$), we remove the trajectories in \mathcal{S}_L that satisfy ϕ^1 and find another pattern ϕ^2 from the remaining trajectories. And the same procedure continues until all the obtained patterns already cover p_{th} portion of the trajectories in \mathcal{S}_L . The formulas ϕ^j (representing different patterns) are then connected in disjunction to form the final inferred formula.

The proposed information-guided temporal logic inference approach is outlined in Algorithm 3. Initially, \mathcal{S}_L is given as the dataset $\{\hat{s}_{1:L}^1, \dots, \hat{s}_{1:L}^m\}$, $\hat{\mathcal{B}}_L$ is given as \emptyset , ϕ is given as \top , and β is given as 0. We define a *primitive template* as a primitive pLTL formula with fixed temporal operator, fixed predicate structure and undetermined parameters (including the temporal parameters and the parameters in the predicates). We select a set \mathcal{P} of primitive templates, and for each primitive template $\phi_k \in \mathcal{P}$ and the parameter vector θ_k , we solve the following optimization problem:

$$\max_{\theta_k \in \Theta_k} \mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_k(\theta_k)}) \text{ subject to } \bar{\mathbb{P}}_{\mathcal{S}_L, \phi_k(\theta_k)} \geq \hat{p}_{\text{th}}, \quad (6)$$

where Θ_k is the set of parameter vectors for ϕ_k , $\hat{p}_{\text{th}} \in (0, p_{\text{th}}]$ is a hyperparameter represent-

ing a coverage threshold for each pattern. \hat{p}_{th} should not be set too large (e.g. $\hat{p}_{\text{th}} = p_{\text{th}}$) as each pattern may not cover a large portion of the trajectories in \mathcal{S}_L . \hat{p}_{th} should not be set too small as well, as it may lead to a very long formula consisting of many patterns in the end and violating the size constraint.

To solve the constrained non-convex optimization problem (6), we use particle swarm optimization (PSO) [13] to optimize the parameter vector θ_k for each primitive template ϕ_k for the following unconstrained non-convex optimization problem transformed from (6):

$$\min_{\theta_k \in \Theta_k} -\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_k(\theta_k)}) + G(\bar{\mathbb{P}}_{\mathcal{S}_L, \phi_k(\theta_k)}), \quad (7)$$

where

$$G(x) = \begin{cases} \varrho \cdot (\hat{p}_{\text{th}} - x) & \text{if } x \leq \hat{p}_{\text{th}}; \\ 0 & \text{otherwise,} \end{cases}$$

and ϱ is a large positive number. As $\phi_k(\theta_k)$ does not contain Boolean connectives, $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_k(\theta_k)})$ can be efficiently computed by Algorithm 1 or Algorithm 2.

After we obtain the formula ϕ_k^* by solving (7) for each primitive template, we rank ϕ_k^* according to $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_k^*})$ (from the highest to the lowest) and obtain $\{\hat{\phi}_j^*\}_{j=1}^{|\mathcal{P}|}$. For the formula $\hat{\phi}_1^*$, we further increase the information gain by adding conjunctions between $\hat{\phi}_1^*$ and the next formula $\hat{\phi}_j^*$ (with j selected in increasing order from 2, 3, \dots , $|\mathcal{P}|$) that satisfies two conditions: (1) $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\hat{\phi}_1^* \wedge \hat{\phi}_j^*}) \geq \alpha \mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\hat{\phi}_1^*})$; (2) $\bar{\mathbb{P}}_{\mathcal{S}_L, \hat{\phi}_1^* \wedge \hat{\phi}_j^*} \geq \hat{p}_{\text{th}}$. Condition (1) guarantees that information gain is increased by adding each conjunction, with larger α ($\alpha > 1$) representing higher information gain required for adding each conjunction, condition (2) guarantees that the coverage threshold we set is still valid for this pattern by adding each conjunction.

To reduce the computational cost in computing the information gain $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ when ϕ contains Boolean connectives, we simulate a set of trajectories $\hat{\mathcal{B}}_L$ based on the prior probability distribution \mathcal{F}_L and calculate an estimated information gain $\hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ using the estimated probability $\mathbb{P}_{\hat{\mathcal{B}}_L, \phi}$ for checking condition (1). If a formula $\hat{\phi}_j^*$ satisfies the two conditions, then we add conjunctions between $\hat{\phi}_1^* \wedge \hat{\phi}_j^*$ and the next formula $\hat{\phi}_k^*$ that satisfies $\hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\hat{\phi}_1^* \wedge \hat{\phi}_j^* \wedge \hat{\phi}_k^*}) \geq \alpha \hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\hat{\phi}_1^* \wedge \hat{\phi}_j^*})$ and $\bar{\mathbb{P}}_{\mathcal{S}_L, \hat{\phi}_1^* \wedge \hat{\phi}_j^* \wedge \hat{\phi}_k^*} \geq \hat{p}_{\text{th}}$, and the same procedure continues until the last formula $\hat{\phi}_{|\mathcal{P}|}^*$ is checked.

After we obtain the formula $\hat{\phi} = \hat{\phi}_1^* \wedge \hat{\phi}_j^* \wedge \dots \wedge \hat{\phi}_q^*$, if the coverage constraint is already satisfied, then the algorithm terminates. Otherwise, we remove the trajectories that satisfy $\hat{\phi}$ and repeat the process for the remaining trajectories in $\mathcal{S}_L^{-\hat{\phi}}$, where $\mathcal{S}_L^{-\hat{\phi}}$ denotes the set of paths in \mathcal{S}_L that do not satisfy $\hat{\phi}$. The obtained formula from $\mathcal{S}_L^{-\hat{\phi}}$ is connected in

Algorithm 3 Information-Guided Temporal Logic Inference.

```

1: procedure  $TLprocedure(\mathcal{S}_L, \hat{\mathcal{B}}_L, \phi, \beta)$ 
2: for  $k = 1 : |\mathcal{P}|$  do
3:    $\phi_k^* \leftarrow \arg \max_{\theta_k \in \Theta_k} \mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_k(\theta_k)})$ 
4: end for
5: Obtain  $\{\hat{\phi}_j^*\}_{j=1}^{|\mathcal{P}|}$  by ranking  $\phi_k^*$  according to  $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_k^*})$ 
6:  $\hat{\phi} \leftarrow \hat{\phi}_1^*$ 
7: if  $\hat{\mathcal{B}}_L = \emptyset$  then
8:   Generate/simulate a set of trajectories  $\hat{\mathcal{B}}_L$  based on the prior probability distribution  $\mathcal{F}_L$ 
9: end if
10: for  $j = 2 : |\mathcal{P}|$  do
11:   if  $\hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\hat{\phi} \wedge \hat{\phi}_j^*}) \geq \alpha \hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\hat{\phi}})$  and  $|\bar{\mathbb{P}}_{\mathcal{S}_L, \hat{\phi} \wedge \hat{\phi}_j^*} - \bar{\mathbb{P}}_{\mathcal{S}_L, \hat{\phi}}| \leq \epsilon$  then  $\hat{\phi} \leftarrow \hat{\phi} \wedge \hat{\phi}_j^*$ 
12:   end if
13: end for
14:  $\beta \leftarrow \beta + (1 - \beta)\bar{\mathbb{P}}_{\mathcal{S}_L, \hat{\phi}}$ ,  $\phi \leftarrow \phi \vee \hat{\phi}$ 
15: if  $\beta \geq p_{th}$  then return  $\phi$ 
16: else  $\mathcal{S}_L^{-\hat{\phi}} \leftarrow \{\hat{s}_{1:L}^i \mid \hat{s}_{1:L}^i \in \mathcal{S}_L, \hat{s}_{1:L}^i \not\models \hat{\phi}\}$ 
17:    $\phi \leftarrow TLprocedure(\mathcal{S}_L^{-\hat{\phi}}, \hat{\mathcal{B}}_L, \phi, \beta)$ 
18: end if
19: Return  $\phi$ 
20: end procedure

```

disjunction with $\hat{\phi}$ and the same procedure goes on until the coverage constraint is satisfied.

The time complexity of Algorithm 3 is $O(M|\mathcal{P}|g(m))$, where $g(m)$ is the time complexity of the local PSO algorithm for each primitive template, m is the number of trajectories in the dataset, M is the number of iterations needed for Algorithm 3 to terminate ($M \leq \log_{1-\hat{p}_{th}}(1 - p_{th}) + 1$). It can be seen that Algorithm 3 has polynomial time complexity with respect to the size of the formula. If we set $(|\mathcal{P}| - 1) \log_{1-\hat{p}_{th}}(1 - p_{th}) \leq \ell_{th}$, then the size constraint in Problem 1 can be guaranteed.

6 Implementation

6.1 Case I: Explaining Anomalous Patterns

In the first case study, we implement the inference approach in identifying and explaining the anomalous patterns “hidden” in a set of trajectories. The *prior* DTMC is as shown in Fig. 3, and it is parametrized with the following parameters: $p_0 = p_{-1} = p_1 = \frac{1}{3}$, $\hat{p}_0 = \hat{p}_1 = \hat{p}_{-1} = 0.5$. The initial probability distribution is set as $p_{int}(s^i) = 0.1$ where $s^i = i$, ($i = 1, 2, \dots, 10$). We inject two anomalous patterns by adding the following constraints when simulating 100 trajectories from the *prior* DTMC: for each trajectory of the first [60](#)

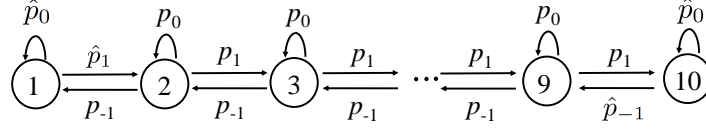


Figure 3: The *prior* DTMC \mathcal{M} in Case I and II.

trajectories, the minimal number for the first 5 time indices are constrained to the number 1 or 2, and the two numbers from time index 51 to 52 are constrained to the number 9 or 10, for each trajectory of the last 60 trajectories, the minimal number during the 10 time indices from 71 to 80 are constrained to the number 1 or 2 (for 20 trajectories the two anomalous patterns both exist). We set $p_{\text{th}} = 0.9, \hat{p}_{\text{th}} = 0.5, \ell_{\text{th}} = 10, \alpha = 1.5, \varrho = 1000$. We infer a pLTL formula from a set of primitive templates [6]

$$\mathcal{P} = \{\Box_I \pi, \Diamond_I \pi, \Box \Diamond_I \pi, \Diamond \Box_I \pi\}, \quad (8)$$

where I can be $\geq i, \leq i$ or $(\geq i_1, \leq i_2)$ (here $i_1 < i_2, i, i_1, i_2 \in \mathbb{T}$), π is an atomic predicate in the form of $x \geq a$ or $x \leq a$, where $a \in [1, 10]$ is an integer parameter. In the first iteration, the primitive pLTL formula with the highest information gain is $\phi_1^{1*} = \Box_{\geq 51, \leq 52}(x \geq 9)$ (see Tab. 1). To add conjunctions with other inferred primitive pLTL formulas, we simulate a set of 100 trajectories from the *prior* DTMC and find only $\phi_4^{1*} = \Diamond_{\leq 5}(x \leq 2)$ satisfying the conditions in line 11 of Algorithm 3 to be added in conjunction with ϕ_1^{1*} . In comparison, ϕ_3^{1*} also has high information gain as a primitive pLTL formula, but the estimated information gain $\hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_1^{1*} \wedge \phi_3^{1*}})$ is computed the same as $\hat{\mathcal{I}}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi_1^{1*}})$, as ϕ_3^{1*} essentially describes the same subpattern as ϕ_1^{1*} . As $\bar{\mathbb{P}}_{\mathcal{S}_L, \phi_1^{1*} \wedge \phi_4^{1*}} = 0.61 < p_{\text{th}}$, the algorithm continues for the second iteration and finds the second pattern $\phi_4^{2*} = \Diamond_{\geq 71, \leq 80}(x \leq 2)$ from the remaining 39 trajectories that do not satisfy $\phi_1^{1*} \wedge \phi_4^{1*}$. Then the algorithm terminates with $\bar{\mathbb{P}}_{\mathcal{S}_L, (\phi_1^{1*} \wedge \phi_4^{1*}) \vee \phi_4^{2*}} = 0.61 + 0.39 \cdot 1 > p_{\text{th}}$. Therefore, the final inferred pLTL formula is

$$\phi^* = (\Box_{\geq 51, \leq 52}(x \geq 9) \wedge \Diamond_{\leq 5}(x \leq 2)) \vee (\Diamond_{\geq 71, \leq 80}(x \leq 2)).$$

6.2 Case II: Explaining Pattern Changes

In the second case study, we implement the inference approach in inferring a pLTL formula to explain the pattern changes from the *prior* DTMC to the DTMC *at present*. We simulate 100 trajectories from the DTMC *at present* as shown in Fig. 3 with the following parameters: $p_0 = 0.2, p_{-1} = 0.6, p_1 = 0.2, \hat{p}_0 = \hat{p}_1 = \hat{p}_{-1} = 0.5$. The *prior* DTMC has the same structure as that of the DTMC *at present*, but with different parameters. In Scenario (a), we set $p_0^a = p_{-1}^a = p_1^a = \frac{1}{3}, \hat{p}_0^a = \hat{p}_1^a = \hat{p}_{-1}^a = 0.5$; in Scenario (b), we set $p_0^b = 0.05, p_{-1}^b = 0.9, p_1^b =$

Table 1: Results in First Iteration of Case Study I

pLTL formula	$\bar{\mathbb{P}}_{\mathcal{S}_L, \phi^*}$	$\mathbb{P}_{\mathcal{B}_L, \phi^*}$	$\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\phi^*})$
$\phi_1^{1*} = \Box_{\geq 51, \leq 52}(x \geq 9)$	0.63	0.1429	0.0062
$\phi_2^{1*} = \Box_{\geq 3, \leq 4}(x \leq 3)$	0.69	0.2609	0.004
$\phi_3^{1*} = \Diamond_{\geq 50, \leq 51}(x \geq 9)$	0.67	0.2143	0.0048
$\phi_4^{1*} = \Diamond_{\leq 5}(x \leq 2)$	0.76	0.2877	0.0048
$\phi_5^{1*} = \Box_{\leq 51}(x \geq 10)$	0.59	0.2906	0.0019
$\phi_6^{1*} = \Box_{\leq 96}(x \leq 2)$	0.99	0.7781	0.0021
$\phi_7^{1*} = \Diamond_{\leq 2}(x \geq 9)$	0.85	0.7167	0.000496
$\phi_8^{1*} = \Diamond_{\leq 3}(x \leq 3)$	0.99	0.7808	0.002

$0.05, \hat{p}_0^b = \hat{p}_1^b = \hat{p}_{-1}^b = 0.5$. In both scenarios, we use the same hyperparameters and the same set of primitive templates as in Case Study I.

In Scenario (a), the final inferred pLTL formula is

$$\begin{aligned} \phi_a^* = & (\Box_{\leq 22}(x \leq 1) \wedge \Diamond_{\leq 78}(x \leq 5)) \vee \\ & (\Box_{\geq 53, \leq 99}(x \leq 4) \wedge \Box_{\leq 32}(x \leq 1)). \end{aligned}$$

In Scenario (b), the final inferred pLTL formula is

$$\phi_b^* = (\Diamond_{\geq 16, \leq 99}(x \geq 5)) \vee (\Diamond_{\geq 13, \leq 99}(x \geq 4)) \vee (\Diamond_{\geq 14, \leq 30}(x \geq 3)).$$

It can be seen that the predicates in ϕ_a^* are all in the form of $x \leq a$, as the DTMC *at present* has the trend towards smaller numbers in comparison with the *prior* DTMC in Scenario (a), while the predicates in ϕ_b^* are all in the form of $x \geq a$, as the *prior* DTMC in Scenario (b) has the trend towards even smaller numbers in comparison with the DTMC *at present*.

6.3 Case III: Explaining Policies of MDPs

In the third case study, we implement the inference approach on a “cops and robbers” game with the inferred pLTL formulas aiming to explain the policies of the robber as an agent whose behaviors are governed by an MDP. As shown in Fig. 4, there are three banks locating in three 1×1 blocks in the state space, where the robber’s goal is to reach *bank*₂ while deceiving the cop (as an adversary) to believe that the robber is trying to reach the other two banks (see [14] for details of the game). We simulate 192 trajectories with length 100, with the policies obtained from [14], starting from the 64 different initial states with 3 different initial beliefs of the cop. We set the prior probability distribution as a stationary uniform distribution in the state space.

In the first task, we infer a pLTL formula from the set of primitive templates (8) by replacing π with ι , where ι is any rectangular predicate in the state space. We use the

Table 2: Results in First Iteration of Second Task of Case Study III

inferred pLTL formula	$\bar{\mathbb{P}}_{\mathcal{S}_L, \psi^*}$	$\mathbb{P}_{\mathcal{B}_L, \psi^*}$	$\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\psi^*})$	$\eta(\psi^*)$
$\psi_{b1,1}^* = \Box(((4 \leq x \leq 6) \wedge (7 \leq y \leq 8) \wedge (b_{\text{cop}} = \text{bank}_1)) \rightarrow \Box_{\leq 5}((4 \leq x \leq 5) \wedge (6 \leq y \leq 7)))$	0.9844	$1.69 \cdot 10^{-5}$	0.1073	0.0007
$\psi_{b1,2}^* = \Box(((3 \leq x \leq 6) \wedge (5 \leq y \leq 8) \wedge (b_{\text{cop}} = \text{bank}_1)) \rightarrow \Diamond_{\leq 2}((4 \leq x \leq 5) \wedge (6 \leq y \leq 7)))$	0.8438	$5.58 \cdot 10^{-16}$	0.292	0.9771
$\psi_{b2,1}^* = \Box(((3 \leq x \leq 4) \wedge (5 \leq y \leq 8) \wedge (b_{\text{cop}} = \text{bank}_2)) \rightarrow \Box_{\leq 5}((3 \leq x \leq 4) \wedge (7 \leq y \leq 8)))$	0.9323	$5.58 \cdot 10^{-16}$	0.325	0.0105
$\psi_{b2,2}^* = \Box(((3 \leq x \leq 6) \wedge (5 \leq y \leq 8) \wedge (b_{\text{cop}} = \text{bank}_2)) \rightarrow \Diamond_{\leq 2}((4 \leq x \leq 6) \wedge (5 \leq y \leq 6)))$	0.9323	$7.8 \cdot 10^{-16}$	0.3218	0.9784
$\psi_{b3,1}^* = \Box(((4 \leq x \leq 6) \wedge (7 \leq y \leq 8) \wedge (b_{\text{cop}} = \text{bank}_3)) \rightarrow \Box_{\leq 4}((4 \leq x \leq 5) \wedge (6 \leq y \leq 7)))$	0.9948	$1.69 \cdot 10^{-5}$	0.109	0.0029
$\psi_{b3,2}^* = \Box(((3 \leq x \leq 6) \wedge (5 \leq y \leq 8) \wedge (b_{\text{cop}} = \text{bank}_3)) \rightarrow \Diamond_{\leq 2}((4 \leq x \leq 5) \wedge (6 \leq y \leq 7)))$	0.9948	$5.58 \cdot 10^{-16}$	0.3491	0.86

same hyperparameters as in Case Study I and II except that p_{th} is set as 0.95 as we expect the explanations of the policies to cover a higher percentage of the trajectories. The final inferred formula for the first task is

$$\phi_f^* = \Diamond \Box_{\leq 30}((3 \leq x \leq 6) \wedge (5 \leq y \leq 8)),$$

which means “the robber will eventually reach the yellow region in Fig. 4 and stay there for at least 30 time indices”.

In the second task, to further explain the policies of the robber as a response to both the current position of the robber and the cop’s beliefs, we infer a pLTL formula in the causal form $\psi = \Box(\psi_c \Rightarrow \psi_e)$. The cause formula is in the form of $\psi_c = \iota_y \wedge \pi_b$, where ι_y is a rectangular predicate of the robber’s position in the yellow region (we shrink the state space to the yellow region as ϕ_f^* indicates) and π_b is a predicate of the belief of the cop in the form of $b_{\text{cop}} = \text{bank}_i$ ($i = 1, 2, 3$). The effect formula ψ_e is in the form of $\psi_e = \Box_I \iota_y$ or $\psi_e = \Diamond_I \iota_y$, where I is defined in (8).

There is a caveat though for pLTL formulas of the causal form $\psi = \Box(\psi_c \Rightarrow \psi_e)$: ψ is satisfied by a trajectory if ψ_c is never satisfied by the trajectory at any time index. To counteract this, we define the truth factor $\eta(\psi)$ as the proportion of the states in all the trajectories that satisfy the cause formula ψ_c , and add $\eta(\psi) > 0$ as an extra constraint in the optimization. Besides, we modify Algorithm 3 for pLTL formulas of the causal form as follows: (i) in line 5 of Algorithm 3, we rank the obtained primitive formulas based on $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\psi_k^*}) + \eta(\psi_k^*)$ instead of $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^{\psi_k^*})$; (ii) in line 16-17 of Algorithm 3, if the inferred formula $\hat{\psi} = \Box(\hat{\psi}_c \Rightarrow \hat{\psi}_e)$ does not satisfy the coverage constraint, then we infer another primitive formula $\hat{\psi}_e^2$ that is connected in disjunction with $\hat{\psi}_e$, and the same procedure continues until the coverage constraint is satisfied. We use the same hyperparameters as in

the first task and infer three pLTL formulas corresponding to the three different beliefs of the cop. After the first iteration, the inferred pLTL formulas for the three different beliefs of the cop are $\psi_{b1,2}^*$, $\psi_{b2,2}^*$ and $\psi_{b3,2}^*$, as shown in Tab. 2. As $\psi_{b3,2}^*$ already satisfies $\bar{\mathbb{P}}_{S_L, \psi_{b3,2}^*} \geq p_{th}$, so the final inferred formula for the belief $b_{cop} = bank_3$ is $\psi_{b3}^* = \psi_{b3,2}^*$. For the other two beliefs, after another iteration, the final inferred formulas are as follows:

$$\psi_{b1}^* = \Box \left(((3 \leq x \leq 6) \wedge (5 \leq y \leq 8) \wedge (b_{cop} = bank_1)) \rightarrow (\Diamond_{\leq 2}((4 \leq x \leq 5) \wedge (6 \leq y \leq 7)) \vee \Diamond_{\leq 2}((5 \leq x \leq 6) \wedge (5 \leq y \leq 6))) \right),$$

$$\psi_{b2}^* = \Box \left(((3 \leq x \leq 6) \wedge (5 \leq y \leq 8) \wedge (b_{cop} = bank_2)) \rightarrow (\Diamond_{\leq 2}((4 \leq x \leq 6) \wedge (5 \leq y \leq 6)) \vee \Diamond_{\leq 2}((3 \leq x \leq 4) \wedge (7 \leq y \leq 8))) \right).$$

ψ_{b1}^* , ψ_{b2}^* and ψ_{b3}^* mean the following: if the robber is in the yellow region and the cop believes that the robber tries to reach $bank_1$, then the robber will eventually go to $Region_2$ or $Region_3$ in 2 time steps; if the robber is in the yellow region and the cop believes that the robber tries to reach $bank_2$, then the robber will go to $Region_3$ or $Region_4$ or $Region_1$ in 2 time steps; if the robber is in the yellow region and the cop believes that the robber tries to reach $bank_3$, then the robber will eventually go to $Region_2$ in 2 time steps.

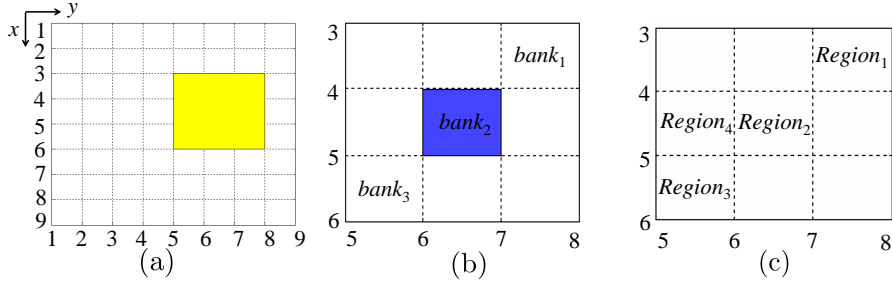


Figure 4: The game setup and inferred regions in Case III.

7 Conclusion

We have proposed an approach to extract interpretable and informative knowledge in the form of pLTL formulas from data. For future work, the computational methods of the information gain can be developed for more general forms of temporal logic formulas, and with other types of prior probability distributions. We will also consider other methods to solve the inference problem that is either more computationally efficient or requiring fewer data.

APPENDIX

Proof of Proposition 1:

From Definition 3 and Definition 4, we have

$$\begin{aligned}
\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi) &= \frac{D_{\text{KL}}(\bar{\mathcal{F}}_L^\phi || \mathcal{F}_L)}{L} \\
&= \frac{1}{L} \sum_{s_{1:L} \in \mathcal{B}_L, s_{1:L} \models \phi} \bar{\mathcal{F}}_L^\phi(s_{1:L}) \log \frac{\bar{\mathcal{F}}_L^\phi(s_{1:L})}{\mathcal{F}_L(s_{1:L})} \\
&\quad + \frac{1}{L} \sum_{s_{1:L} \in \mathcal{B}_L, s_{1:L} \not\models \phi} \bar{\mathcal{F}}_L^\phi(s_{1:L}) \log \frac{\bar{\mathcal{F}}_L^\phi(s_{1:L})}{\mathcal{F}_L(s_{1:L})} \\
&= \frac{1}{L} \sum_{s_{1:L} \in \mathcal{B}_L, s_{1:L} \models \phi} \frac{\mathcal{F}_L(s_{1:L}) \bar{\mathbb{P}}_{\mathcal{B}_L, \phi}}{\mathbb{P}_{\mathcal{B}_L, \phi}} \log \frac{\bar{\mathbb{P}}_{\mathcal{B}_L, \phi}}{\mathbb{P}_{\mathcal{B}_L, \phi}} \\
&\quad + \frac{1}{L} \sum_{s_{1:L} \in \mathcal{B}_L, s_{1:L} \not\models \phi} \frac{\mathcal{F}_L(s_{1:L}) \bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi}}{\mathbb{P}_{\mathcal{B}_L, \neg \phi}} \log \frac{\bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi}}{\mathbb{P}_{\mathcal{B}_L, \neg \phi}}.
\end{aligned}$$

As $\sum_{s_{1:L} \in \mathcal{B}_L, s_{1:L} \models \phi} \mathcal{F}_L(s_{1:L}) = \mathbb{P}_{\mathcal{B}_L, \phi}$, we have

$$\begin{aligned}
\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi) &= \frac{\bar{\mathbb{P}}_{\mathcal{B}_L, \phi}}{L} \log \frac{\bar{\mathbb{P}}_{\mathcal{B}_L, \phi}}{\mathbb{P}_{\mathcal{B}_L, \phi}} + \frac{\bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi}}{L} \log \frac{\bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi}}{\mathbb{P}_{\mathcal{B}_L, \neg \phi}} \\
&= \frac{1}{L} (\bar{\mathbb{P}}_{\mathcal{B}_L, \phi} \log(\bar{\mathbb{P}}_{\mathcal{B}_L, \phi}) - \bar{\mathbb{P}}_{\mathcal{B}_L, \phi} \log(\mathbb{P}_{\mathcal{B}_L, \phi}) \\
&\quad + \bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi} \log(\bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi}) - \bar{\mathbb{P}}_{\mathcal{B}_L, \neg \phi} \log(\mathbb{P}_{\mathcal{B}_L, \neg \phi})).
\end{aligned}$$

Proof of Proposition 2:

For a syntactically co-safe pLTL formula ϕ , we use $p_L^\phi(\ell, q_k)$ ($\ell \leq L$) to denote the probability of extending a trajectory of length ℓ to length L , with the state of the DFA \mathcal{A}^ϕ at time index ℓ being the state q_k , such that the extended trajectory of length L strongly satisfies ϕ . Then we have

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

We can compute $p_L^\phi(\ell, q_k)$ recursively as follows:

$$\begin{bmatrix} p_L^\phi(\ell-1, q_0) \\ \vdots \\ p_L^\phi(\ell-1, q_K) \end{bmatrix} = \begin{bmatrix} c_{0,0} & c_{0,1} & \dots & c_{0,K} \\ \vdots & \vdots & \vdots & \vdots \\ c_{K,0} & c_{K,1} & \dots & c_{K,K} \end{bmatrix} \begin{bmatrix} p_L^\phi(\ell, q_0) \\ \vdots \\ p_L^\phi(\ell, q_K) \end{bmatrix}, \quad (9)$$

where $c_{j,k}$ is the probability of transitioning from q_j to q_k (which can be calculated from \mathcal{F}_L). Finally, we have $\mathbb{P}_{\mathcal{B}_L, \phi} = p_L^\phi(0, q_0)$.

For a syntactically safe pLTL formula ϕ , we use $p_L^{-\phi}(\ell; q_k)$ ($\ell \leq L$) to denote the probability of extending a path of time length ℓ to time length L , with the state of the DFA $\mathcal{A}^{-\phi}$ at time instant ℓ being the state q_k , such that the extended path of time length L strongly violates ϕ (i.e. strongly satisfies $\neg\phi$). Then we have the following:

$$p_L^{-\phi}(L; q_k) = \begin{cases} 1 & \text{if } q_k \in \text{Acc}^{-\phi}; \\ 0 & \text{otherwise;} \end{cases}$$

We can compute $p_L^{-\phi}(\ell; q_k)$ recursively using (9) by replacing each ϕ with $\neg\phi$. Finally, we have $\mathbb{P}_{\mathcal{B}_L, \phi} = 1 - p_L^{-\phi}(0; q_0)$.

With $\mathbb{P}_{\mathcal{B}_L, \phi}$, we can calculate $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ according to Proposition 1.

Proof of Proposition 3:

For a syntactically co-safe pLTL formula ϕ , we use $p_L^\phi(\ell, s^i, q_k)$ to denote the probability of extending a trajectory of length ℓ to length L , with the state of the product automaton \mathcal{M}_p^ϕ (of \mathcal{M} and \mathcal{A}^ϕ) at time index ℓ being the state (s^i, q_k) , such that the extended trajectory of length L strongly satisfies ϕ . Then we have the following:

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

We can compute $p_L^\phi(\ell, s^i, q_k)$ recursively as follows:

$$\begin{bmatrix} \mathbf{p}_L^\phi(\ell-1, \cdot, q_0) \\ \vdots \\ \mathbf{p}_L^\phi(\ell-1, \cdot, q_K) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{0,0}^p & \cdots & \mathbf{C}_{0,K}^p \\ \vdots & \vdots & \vdots \\ \mathbf{C}_{K,0}^p & \cdots & \mathbf{C}_{K,K}^p \end{bmatrix} \begin{bmatrix} \mathbf{p}_L^\phi(\ell, \cdot, q_0) \\ \vdots \\ \mathbf{p}_L^\phi(\ell, \cdot, q_K) \end{bmatrix} \quad (10)$$

where $\mathbf{C}_{j,k}^p(i_1, i_2) = P_p^\phi((s^{i_1}, q_j), (s^{i_2}, q_k))$, $\mathbf{p}_L^\phi(\ell, \cdot, q_k) := [p_L^\phi(\ell, s^1, q_k), p_L^\phi(\ell, s^2, q_k), \dots, p_L^\phi(\ell, s^H, q_k)]^T$.

Finally, we have $\mathbb{P}_{\mathcal{B}_L, \phi} = \sum_{s^i \in S_0} p_{\text{int}}(s^i) p_L^\phi(0, s^i, q_0)$, where $p_{\text{int}}(s^i)$ is the probability of the initial state being s^i .

For a syntactically safe pLTL formula ϕ , we use $p_L^{-\phi}(\ell; s^i, q_k)$ to denote the probability of extending a path of time length ℓ to time length L , with the state of the product automaton $\mathcal{M}_L^{-\phi}$ (of \mathcal{M} and $\mathcal{A}_L^{-\phi}$) at time instant ℓ being the state (s^i, q_k) , such that the extended path of time length L strongly violates ϕ (i.e. strongly satisfies $\neg\phi$). Then we have the following:

$$p_L^{-\phi}(L; s^i, q_j) = \begin{cases} 1 & \text{if } q_j \in \text{Acc}^{-\phi}; \\ 0 & \text{otherwise.} \end{cases}$$

We can compute $p_L^{-\phi}(\ell; s^i, q_k)$ recursively using (10) by replacing each ϕ with $\neg\phi$. Finally, we have $\mathbb{P}_{\mathcal{B}_L, \phi} = \sum_{s^i \in S_0} p_{\text{int}}(s^i)(1 - p_L^{-\phi}(0; s^i, q_0))$.

With $\mathbb{P}_{\mathcal{B}_L, \phi}$, we can calculate $\mathcal{I}(\mathcal{F}_L, \bar{\mathcal{F}}_L^\phi)$ according to Proposition 1.

References

- [1] 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.
- [2] H. Yang, B. Hoxha, and G. Fainekos, “Querying parametric temporal logic properties on embedded systems,” in *Proc. Int. Conf. Testing Soft. and Syst.*, vol. 7641 LNCS, 2012, pp. 136–151.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] D. Neider and I. Gavran, “Learning linear temporal properties.” [Online]. Available: <https://arxiv.org/abs/1806.03953>
- [8] S. Chakraborty and J.-P. Katoen, “Parametric LTL on Markov chains,” in *Theoretical Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 207–221.
- [9] R. Alur, K. Etessami, S. La Torre, and D. Peled, “Parametric temporal logic for “Model Measuring”,” *ACM Trans. Comput. Logic*, vol. 2, no. 3, pp. 388–407, Jul. 2001.
- [10] H.-M. Ho, J. Ouaknine, and J. Worrell, “Online monitoring of metric temporal logic,” in *Proc. Int. Conf. Runtime Verification*, B. Bonakdarpour and S. A. Smolka, Eds. Cham: Springer Int. Publishing, 2014, pp. 178–192.

- [11] O. Kupferman and M. Y. Vardi, “Model checking of safety properties,” *Form. Methods Syst. Des.*, vol. 19, no. 3, pp. 291–314, Oct. 2001. [Online]. Available: <https://doi.org/10.1023/A:1011254632723>
- [12] C. Belta, B. Yordanov, and E. A. Gol, *Formal Methods for Discrete-Time Dynamical Systems*. Springer, 2016.
- [13] F. Zhang, J. Cao, and Z. Xu, “An improved particle swarm optimization particle filtering algorithm,” in *Proc. Int. Conf. Commun., Circuits and Syst.*, vol. 2, 2013, pp. 173–177.
- [14] M. Ornik and U. Topcu, “Deception in optimal control,” accepted to 56th Annual Allerton Conf. Communication, Control and Computing, 2018. [Online]. Available: [extendedversionathttps://arxiv.org/abs/1805.03090](https://arxiv.org/abs/1805.03090)