End-to-end Learning of LTL_f Formulae by Faithful LTL_f Encoding

Hai Wan[1], Pingjia Liang[1], Jianfeng Du[2,3], Weilin Luo[1], Rongzhen Ye[1], Bo Peng[1]

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, P.R.China
[2]Guangdong University of Foreign Studies, Guangzhou 510006, P.R.China
[3]Bigmath Technology, Shenzhen 518063, P.R.China

Content

Content

Definition of Problem

Learning formulae to characterize the high-level behavior of a system from observation traces.

- focus on the *linear temporal logic on finite traces* (LTL_f) formula
- arbitrary form
- noisy data

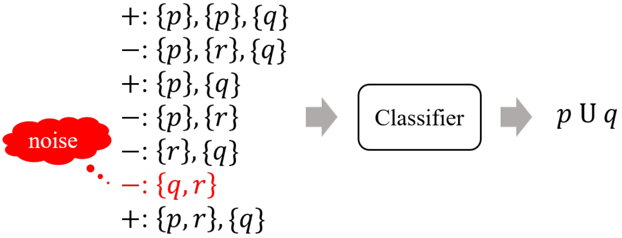

Figure 1: Learning LTL_f formulae from imperfect data.

Significance and Challenge

Wide applications:

1. verification of system properties [3]
2. behavior classification [1]
3. explainable models [4]

Significance and Challenge

Wide applications:

1. verification of system properties [3]
2. behavior classification [1]
3. explainable models [4]

Challenging task:

1. *huge search space* of the target formula in arbitrary form
2. *wrong search bias* resulting from noisy data

State-of-the-art Approach

State-of-the-art (SOTA) approach to learn LTL_f formulae:

- SAT-based [1,6]
- based on bayesian inference [4]

They either assume a noise-free environment or restrict the hypothesis space by LTL_f templates.

State-of-the-art Approach

State-of-the-art (SOTA) approach to learn LTL_f formulae:

- SAT-based[1,6]
- based on bayesian inference[4]

They either assume a noise-free environment or restrict the hypothesis space by LTL_f templates.

Gaglione et al. (2021)[2]:

- MaxSAT-based approach

The scalability of them is limited in calling the MaxSAT solver.

State-of-the-art Approach

State-of-the-art (SOTA) approach to learn LTL_f formulae:
Luo et al. (2022)[5]:

- GNN-based approach

Significant performance gap between the neural network and the interpreted formula.

State-of-the-art Approach

State-of-the-art (SOTA) approach to learn LTL_f formulae:
Luo et al. (2022)[5]:

- GNN-based approach

Significant performance gap between the neural network and the interpreted formula.

Developing new approaches based on *neural networks* to learn arbitrary LTL_f formulae from imperfect data.

Content

The Parameter Set of TLTLf

Definition 1 (The parameter set of TLTLf)

Let $\mathbb{P}$ be a set of atomic propositions and $L \in \mathbb{N}$. The parameter set of TLTLf of size L is defined as $\Gamma = \{(\Gamma_{\text{right}})_{i,j} \in \mathbb{R} | 1 \leq i \leq L-2, i+2 \leq j \leq L\} \cup \{(\Gamma_{\text{atom}})_{i,j} \in \mathbb{R} | 1 \leq i \leq L, 1 \leq j \leq |\mathbb{P}|\} \cup \{(\Gamma_{\neg})_i, (\Gamma_{\wedge})_i, (\Gamma_{\mathsf{X}})_i, (\Gamma_{\mathsf{U}})_i, (\Gamma_{\text{none}})_i \in \mathbb{R} | 1 \leq i \leq L\}$. For brevity, we also reuse Γ to denote an assignment of the parameter set Γ of TLTLf.

LTL$_f$ Encoding of TLTLf

Definition 2 (LTL$_f$ encoding of TLTLf)

An LTL$_f$ encoding of TLTLf of size L is defined as
$\theta = \{(\theta_{\text{right}})_{i,j} \in \mathbb{R}^{(0,1)} | 1 \leq i \leq L-2, i+2 \leq j \leq L\} \cup \{(\theta_{\text{atom}})_{i,j} \in \mathbb{R}^{(0,1)} | 1 \leq i \leq L, 1 \leq j \leq |\mathbb{P}|\} \cup \{(\theta_{\neg})_i, (\theta_{\wedge})_i, (\theta_{\mathsf{X}})_i, (\theta_{\mathsf{U}})_i, (\theta_{\text{none}})_i \in \mathbb{R}^{(0,1)} | 1 \leq i \leq L\}$, where $\mathbb{R}^{(0,1)}$ denotes the real value range from 0 to 1.

LTL$_f$ Encoding

Example 1

Let $\mathbb{P} = \{p_1, p_2\}$ and θ be an LTL$_f$ encoding of TLTLf of size 3 where $(\theta_\neg)_1 = 0.8, (\theta_\mathsf{X})_1 = 0.3, (\theta_{\text{atom}})_{2,1} = (\theta_{\text{none}})_3 = 1$ and other parameters are assigned 0. The LTL$_f$ formula that θ represents is the most likely to be $\neg p_1$ while it may also be $\mathsf{X}p_1$ since $(\theta_\mathsf{X})_1 = 0.3$.

LTL$_f$ Encoding of TLTLf

In fact, an arbitrary parameter assignment of TLTLf can be converted to an LTL$_f$ encoding of TLTLf of the same size, as shown in the following equation.

$$\begin{aligned}&(\theta_{\text{right}})_{i,j} = \frac{e^{(\Gamma_{\text{right}})_{i,j}}}{(\eta_{\text{right}})_i}, (\theta_{\text{atom}})_{i,j} = \frac{e^{(\Gamma_{\text{atom}})_{i,j}}}{(\eta_{\text{op}})_i},\\ &(\theta_{\neg})_i = \frac{e^{(\Gamma_{\neg})_i}}{(\eta_{\text{op}})_i}, (\theta_{\wedge})_i = \frac{e^{(\Gamma_{\wedge})_i}}{(\eta_{\text{op}})_i}, (\theta_{\mathsf{X}})_i = \frac{e^{(\Gamma_{\mathsf{X}})_i}}{(\eta_{\text{op}})_i}, (\theta_{\mathsf{U}})_i = \frac{e^{(\Gamma_{\mathsf{U}})_i}}{(\eta_{\text{op}})_i}, (\theta_{\text{none}})_i = \frac{e^{(\Gamma_{\text{none}})_i}}{(\eta_{\text{op}})_i},\end{aligned} \tag{1}$$

where

$$\begin{aligned}(\eta_{\text{right}})_i &= \textstyle\sum_{j=i+2}^{L} e^{(\Gamma_{\text{right}})_{i,j}} + \sum_{j=1}^{|\mathbb{P}|} e^{(\Gamma_{\text{atom}})_{i,j}} + e^{(\Gamma_{\text{none}})_i} + e^{(\Gamma_{\neg})_i} + e^{(\Gamma_{\mathsf{X}})_i},\\ (\eta_{\text{op}})_i &= \textstyle\sum_{j=1}^{|\mathbb{P}|} e^{(\Gamma_{\text{atom}})_{i,j}} + e^{(\Gamma_{\text{none}})_i} + e^{(\Gamma_{\neg})_i} + e^{(\Gamma_{\wedge})_i} + e^{(\Gamma_{\mathsf{X}})_i} + e^{(\Gamma_{\mathsf{U}})_i}.\end{aligned} \tag{2}$$

Inference of TLTLf

Definition 3

Let $\mathbb{P}$ be a set of atomic propositions and Γ a parameter assignment of TLTLf of size $L \in \mathbb{N}$. θ is constructed from Γ by Equation (1). Given a trace $\pi = s_0, s_1, \ldots, s_n$ over $\mathbb{P}$, TLTLf computes satisfaction vectors $x_i \in \mathbb{R}^L$ (where $0 \leq i \leq n$) defined as follows:

$$\begin{aligned}(x_i)_j = \quad & \sigma(\textstyle\sum_{k=1}^{|\mathbb{P}|}(\theta_{\text{atom}})_{j,k} I(p_k \in s_i) + (\theta_{\mathsf{X}})_j (x_{i+1})_{j+1} \\ & +(\theta_{\neg})_j \sigma(1 - (x_i)_{j+1}) + (\theta_{\wedge})_j \sigma((x_i)_{j+1} + (r_i)_j - 1) \\ & +(\theta_{\mathsf{U}})_j ((r_i)_j + \sigma((x_i)_{j+1} + (x_{i+1})_j - 1))),\end{aligned} \tag{3}$$

where

$$\begin{aligned}(r_i)_j \quad & = \textstyle\sum_{k=L}^{j+2} (\theta_{\text{right}})_{j,k} (x_i)_k, \\ \sigma(x) \quad & = \min(1, \max(0, x)),\end{aligned} \tag{4}$$

and $(x_i)_{L+1} = 0$, $(x_{n+1})_k = 0$ for all $1 \leq k \leq L+1$, and $I(C)$ returns 1 if C is satisfied or 0 otherwise. By $\mathrm{ESat}(\theta, \pi)$ we denote the satisfaction relation between θ and π. Finally TLTLf outputs $\mathrm{ESat}(\theta, \pi)$ as $(x_0)_1$.

Inference of TLTLf

Example 2

Let $\pi = \{p_1, p_2\}, \{p_2\}$ and θ be the LTL$_f$ encoding of TLTLf given in Example 1. Then the satisfaction vector is $x_0 = [0, 1, 0], x_1 = [0.8, 0, 0]$. The inference output is $\mathrm{ESat}(\theta, \pi) = (x_0)_1 = 0$.

The Faithful Subclass of LTL$_f$ Encoding

Definition 4 (Faithful LTL$_f$ encoding)

Let θ be an LTL$_f$ encoding of TLTLf of size L. θ is said to be *faithful* if it satisfies the following conditions:

1. $\forall \gamma \in \theta : \gamma = 0 \vee \gamma = 1$.
2. $\forall i \in [1, L] : (\theta_{\text{none}})_i + \sum_{j=1}^{|\mathbb{P}|} (\theta_{\text{atom}})_{i,j} + (\theta_{\neg})_i + (\theta_{\wedge})_i + (\theta_{\mathsf{X}})_i + (\theta_{\mathsf{U}})_i = 1$.
3. $\sum_{j=1}^{|\mathbb{P}|} (\theta_{\text{atom}})_{L,j} + (\theta_{\text{none}})_L = 1 \wedge \forall i \in [1, L-1] :$ $\sum_{j=i+2}^{L} (\theta_{\text{right}})_{i,j} + (\theta_{\text{none}})_i + \sum_{j=1}^{|\mathbb{P}|} (\theta_{\text{atom}})_{i,j} + (\theta_{\neg})_i + (\theta_{\mathsf{X}})_i = 1$.
4. $(\theta_{\text{none}})_1 = 0 \wedge \forall i \in [2, L] :$ $\sum_{j=1}^{i-2} (\theta_{\text{right}})_{j,i} + (\theta_{\neg})_{i-1} + (\theta_{\wedge})_{i-1} + (\theta_{\mathsf{X}})_{i-1} + (\theta_{\mathsf{U}})_{i-1} + (\theta_{\text{none}})_i = 1$.
5. $\forall i \in [1, L-1] : (\theta_{\text{none}})_{i+1} \geq (\theta_{\text{none}})_i$.
6. $\forall i \in [1, L), \forall j \in (i, L], \forall t \in (i, j), \forall t' \in (j, L] : (\theta_{\text{right}})_{i,j} + (\theta_{\text{right}})_{t,t'} \leq 1$.

The Faithful Subclass of LTL$_f$ Encoding

Example 3

Consider the LTL$_f$ encoding θ of TLTLf given in Example 1 again. A faithful LTL$_f$ encoding closed to θ is $\hat{\theta}$, where $(\hat{\theta}_{\neg})_1 = 1, (\hat{\theta}_{\mathsf{X}})_1 = 0, (\hat{\theta}_{\mathrm{atom}})_{2,1} = 1, (\hat{\theta}_{\mathrm{none}})_3 = 1$ and other parameters are assigned 0. The LTL$_f$ formula that $\hat{\theta}$ represents is unique, which is $\neg p_1$.

Faithful LTL$_f$ Encoding vs LTL$_f$ Formula

For an arbitrary LTL$_f$ formula ϕ, we introduce a function to encode ϕ into a parameter assignment of TLTLf of an equal or greater size, formalized in the following Definition 5.

Definition 5

Let ϕ be an LTL$_f$ formula, $T(\phi)$ its syntax tree, and $\text{pretravel}(T(\phi)) = v_1, v_2, ..., v_L$. The function for encoding ϕ into a parameter assignment of TLTLf of size $L' \geq L$, denoted by $\theta_{\phi(L')}$, is defined as follows:

- $\forall 1 \leq i \leq L : (\theta_{\text{right}})_{i,j} = 1$ if v_j is the right child of v_i and $(\theta_{\text{right}})_{i,j} = 0$ otherwise.
- $\forall 1 \leq i \leq L : (\theta_{\text{atom}})_{i,j} = 1$ if $v_i = v_{p_j}$ and $(\theta_{\text{atom}})_{i,j} = 0$ otherwise.
- $\forall 1 \leq i \leq L : (\theta_\beta)_i = 1$ if $v_i = v_\beta$ and $(\theta_\beta)_i = 0$ otherwise, where $\beta \in \{\neg, \wedge, \mathsf{X}, \mathsf{U}\}$.
- $\forall L < i \leq L' : (\theta_{\text{none}})_i = 1, (\theta_{\text{right}})_{i,j} = 0, (\theta_{\text{atom}})_{i,j} = 0, (\theta_\beta)_i = 0$, where $\beta \in \{\neg, \wedge, \mathsf{X}, \mathsf{U}\}$.

Faithful LTL$_f$ Encoding vs LTL$_f$ Formula

Example 4

Let ϕ be $p_1 \mathsf{U} \mathsf{X} p_2$. We have $\mathrm{pretravel}(T(\phi)) = v_{\mathsf{U}}, v_{p_1}, v_{\mathsf{X}}, v_{p_2}$. Then in the encoding $\theta_{\phi(5)}$, we have $(\theta_{\mathsf{U}})_1 = 1$, $(\theta_{\mathrm{atom}})_{2,1} = 1$, $(\theta_{\mathsf{X}})_3 = 1$, $(\theta_{\mathrm{atom}})_{4,2} = 1$, $(\theta_{\mathrm{right}})_{1,3} = 1$, $(\theta_{\mathrm{none}})_5 = 1$, and other parameters are assigned 0.

Lemma 1

Let ϕ be an LTL$_f$ formula, then $\theta_{\phi(L')}$ is a faithful LTL$_f$ encoding of $\mathtt{TLTL}f$ of size L'.

Faithful LTL$_f$ Encoding vs LTL$_f$ Formula

The following Definition 6 shows how to decode a faithful LTL$_f$ encoding to a symbol sequence.

Definition 6

Let θ be a faithful LTL$_f$ encoding of TLTLf of size L. The decoding function $\mathrm{decode}(\theta) = o_1 \ldots o_L$ is defined as follows:

$$o_i = \begin{cases} p_j, & (\theta_{\mathrm{atom}})_{i,j} = 1, \\ \beta, & (\theta_\beta)_i = 1, \beta \in \{\neg, \wedge, \mathsf{X}, \mathsf{U}\}, \\ \epsilon, & (\theta_{\mathrm{none}})_i = 1. \end{cases} \tag{5}$$

Faithful LTL$_f$ Encoding vs LTL$_f$ Formula

Example 5

Consider ϕ and $\theta_{\phi(5)}$ in Example 4 again. Since $(\theta_{\mathsf{U}})_1 = 1, (\theta_{\text{atom}})_{2,1} = 1, (\theta_{\mathsf{X}})_3 = 1, (\theta_{\text{atom}})_{4,2} = 1$ and $(\theta_{\text{none}})_5 = 1$, We have $\text{decode}(\theta_{\phi(5)}) = \mathsf{U}p_1\mathsf{X}p_2$, which is exactly the prefix form of ϕ.

Faithful LTL$_f$ Encoding vs LTL$_f$ Formula

The following Theorem 2 shows that the decoding method always results in the prefix form of an LTL$_f$ formula.

Theorem 2

For every faithful LTL$_f$ encoding θ of $TLTLf$, $\mathrm{decode}(\theta)$ is the prefix form of a certain LTL$_f$ formula.

Faithful LTL$_f$ Encoding vs LTL$_f$ Formula

The decoding method is both subjective (Theorem 3) and injective (Theorem 4), which shows that faithful LTL$_f$ encodings and the prefix forms of LTL$_f$ formulae have one-to-one correspondence

Theorem 3

For any LTL$_f$ formula ϕ with $\mathrm{pretravel}(T(\phi)) = v_1, v_2, ..., v_L$ and any $L' \geq L$, there exists a faithful LTL$_f$ encoding θ of size L' such that $\mathrm{decode}(\theta) = \mathrm{pre}(\phi)$.

Theorem 4

Given two different faithful LTL$_f$ encodings of the same size, namely θ_1 and θ_2, $\mathrm{decode}(\theta_1) \neq \mathrm{decode}(\theta_2)$.

Framework of TLTLf

For learning LTL$_f$ formulae, we first build TLTLf parameterized by an LTL$_f$ encoding and then train it to distinguish positive traces from negative traces. Afterwards, we give an algorithm to extract the formula from TLTLf.

Classification Objective

For each trace π in the set of positive traces Π^+ and the set of negative traces Π^-, we use the LTL$_f$ encoding θ to infer the satisfaction relation. The classification objective is:

$$\zeta_1 = \sum_{\pi \in \Pi} (\mathrm{ESat}(\theta, \pi) - \mathsf{lab}(\pi))^2, \tag{6}$$

where $\forall \pi \in \Pi^+, \mathsf{lab}(\pi) = 1$ and $\forall \pi \in \Pi^-, \mathsf{lab}(\pi) = 0$.

Regularization Terms

The regularization terms are formulated as:

$$\begin{aligned}\zeta_2 &= \sum_{i=2}^{L}\Big(\sum_{j=1}^{i-2}(\theta_{\text{right}})_{j,i} + (\theta_{\neg})_{i-1} + (\theta_{\wedge})_{i-1} \\ &\quad + (\theta_{\mathsf{X}})_{i-1} + (\theta_{\mathsf{U}})_{i-1} + (\theta_{\text{none}})_i - 1\Big)^2, \\ \zeta_3 &= \sum_{i=1}^{L-1}\mathsf{Relu}((\theta_{\text{none}})_i - (\theta_{\text{none}})_{i+1}), \\ \zeta_4 &= \sum_{i=1}^{L-2}\sum_{j=i+2}^{L}\sum_{t=i+1}^{j-1}\sum_{t'=j+1}^{L}\mathsf{Relu}((\theta_{\text{right}})_{i,j} + (\theta_{\text{right}})_{t,t'} - 1).\end{aligned} \tag{7}$$

They are obtained from the corresponding conditions by converting constraints like $x = y$ to $(x-y)^2$ and $x > y$ to $\mathsf{Relu}(y - x)$.

Objective

The final objective to be minimized is:

$$\zeta = \zeta_1 + \alpha_1 \zeta_2 + \alpha_2 \zeta_3 + \alpha_3 \zeta_4, \tag{8}$$

where $\alpha_1, \alpha_2, \alpha_3$ are coefficients for regularization terms.

LTL$_f$ Encoding Interpretation

For any $1 \leq i \leq L$, interpret a sub-formula from θ.

- First recommend a set of candidate formulae based on the product of relevant parameters.
- Then select the best formula based on a *expensive metric*, *i.e.*, the discrimination effect for the traces.

Content

Setting

Benchmarks:

- 5 domains for $k_f \in \{3, 6, 9, 12, 15\}$
- For each domain, 50 datasets
- For each dataset,
  - randomly target formulahas k_f sub-formulae of non-atomic propositions
  - randomly $250/250$ positive/negative traces as the training set
  - randomly $500/500$ positive/negative traces as the test set

Setting

Benchmarks:

- 5 domains for $k_f \in \{3, 6, 9, 12, 15\}$
- For each domain, 50 datasets
- For each dataset,
  - randomly target formulahas k_f sub-formulae of non-atomic propositions
  - randomly $250/250$ positive/negative traces as the training set
  - randomly $500/500$ positive/negative traces as the test set

Benchmarks (imperfect):

- randomly chose some traces from the benchmarks and give them wrong labels
- imperfect rate

Setting

Competitors:

Table 1: Details about SOTA approaches.

approach	imperfect data	arbitrary formulae
C.&M.[1]	×	✓
BayesLTL[4]	✓	×
MaxSAT-DT[2]	✓	✓
GLTLf[5]	✓	✓
TLTLf (Ours)	✓	✓

Setting

Competitors:

Table 1: Details about SOTA approaches.

approach	imperfect data	arbitrary formulae
C.&M. [1]	×	✓
BayesLTL [4]	✓	×
MaxSAT-DT [2]	✓	✓
GLTLf [5]	✓	✓
TLTLf (Ours)	✓	✓

Tasks:

- All approaches first learn an LTL_f formula from the training set and then are compared by evaluating the classification effect of the learned formulae on the test set.

Comparisons across datasets

Table 2: Experimental results for $L = 10$ across different approaches. Acc stands for the average accuracy (%) for successful cases. F_1 stands for the average F_1 score (%) for successful cases. N_s stands for the number of cases out of total 50 cases that are successfully solved within the time limit.

	$k_f = 3$			$k_f = 6$			$k_f = 9$			$k_f = 12$			$k_f = 15$		
	Acc(%)	F_1(%)	N_s	Acc(%)	F_1(%)	N_s	Acc(%)	F_1(%)	N_s	Acc(%)	F_1(%)	N_s	Acc(%)	F_1(%)	N_s
MaxSAT-DT	100	100	49	100	100	19	100	100	8	100	100	5	100	100	5
C.&M.	99.77	99.77	50	97.93	96.79	47	97.14	95.55	35	95.10	91.91	20	93.74	87.49	8
BayesLTL	85.19	85.96	50	77.94	76.78	50	74.08	75.73	50	72.77	73.47	50	74.85	77.32	50
GLTLf	94.34	94.27	50	90.09	90.37	50	84.02	83.29	50	83.08	83.21	50	83.07	83.50	50
TLTLf	98.00	97.92	50	95.38	95.47	50	91.90	91.33	50	89.59	88.98	50	90.40	90.22	50

- TLTLf significantly surpasses BayesLTL and GLTLf.
- Although MaxSAT-DT and C.&M. are in the lead, they cannot solve long formulae.

Comparisons on Imperfect Data

- TLTLf performs better then other approaches on imperfect data.
- TLTLf can handle long formulae with higher performance compared to other approaches.

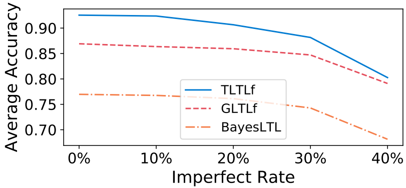

Figure 2: Accuracy achieved by different imperfect rates. The results are averaged by 5 datasets with $k_f \in \{3, 6, 9, 12, 15\}$.

Comparison on the Performance of Interpreting.

- Both TLTLf and GLTLf involve two parts network training and interpreting, so we compare the performance gap between the two parts for TLTLf with that for GLTLf.
- TLTLf has a smaller performance gap than GLTLf.
- This result suggests that the neural model underpinned TLTLf is more interpretable.

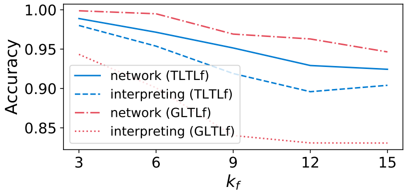

Figure 3: Network accuracy and the accuracy of formula interpreted from the network.

Content

Conclusion and Future Work

Conclusion:

1. We have proposed `TLTLf` parameterized by the LTL_f encoding to simulate LTL_f inference. `TLTLf` bridges the gap between the concise tree-structured syntax and the complex LTL_f semantics.
2. We have identified the faithful LTL_f encoding, which has a one-to-one correspondence to the prefix form of LTL_f formulae.
3. Experiment results demonstrate that `TLTLf` achieves the SOTA performance and yields LTL_f formulae more consistent with the learnt neural network than existing approaches do.

Future work:

1. Future work will extend our approach to LTL or other formal languages.

References I

[1] Camacho, A.; and McIlraith, S. A. 2019. Learning Interpretable Models Expressed in Linear Temporal Logic. In *ICAPS*, 621–630.

[2] Gaglione, J.; Neider, D.; Roy, R.; Topcu, U.; and Xu, Z. 2021. Learning Linear Temporal Properties from Noisy Data: A MaxSAT-Based Approach. In *ATVA*, volume 12971, 74–90.

[3] Kasenberg, D.; and Scheutz, M. 2017. Interpretable apprenticeship learning with temporal logic specifications. In *CDC*, 4914–4921.

[4] Kim, J.; Muise, C.; Shah, A.; Agarwal, S.; and Shah, J. 2019. Bayesian Inference of Linear Temporal Logic Specifications for Contrastive Explanations. In *IJCAI*, 5591–5598.

[5] Luo, W.; Liang, P.; Du, J.; Wan, H.; Peng, B.; and Zhang, D. 2022. Bridging LTLf Inference to GNN Inference for Learning LTLf Formulae. In *AAAI*, 9849–9857.

[6] Neider, D.; and Gavran, I. 2018. Learning Linear Temporal Properties. In *FMCAD*, 1–10.

Thank you for your listening!