Bridging LTL$_f$ Inference to GNN Inference for Learning LTL$_f$ Formulae

Weilin Luo[1], Pingjia Liang[1], Jianfeng Du[2,3,*], Hai Wan[1,*], Bo Peng[1], Delong Zhang[1]
[1]Sun Yat-sen University, Guangzhou, China [2]Guangdong University of Foreign Studies, Guangzhou, China [3]Pazhou Lab, Guangzhou, China

Motivation

Problem. Learning *linear temporal logic on finite traces* (LTL$_f$) formulae in *arbitrary form* to characterize the high-level behavior of a system from observation traces with *noise*.

Challenge.

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

Contributions

1. We theoretically bridge LTL$_f$ inference to GNN inference, which provides a new method for learning arbitrary LTL$_f$ formulae from noisy data.
2. Based on the theoretical result, we design a GNN-based approach, named as `GLTLf`.
3. Experimental results demonstrate that our approach is stronger robustness for noisy data and better scalability in data size.

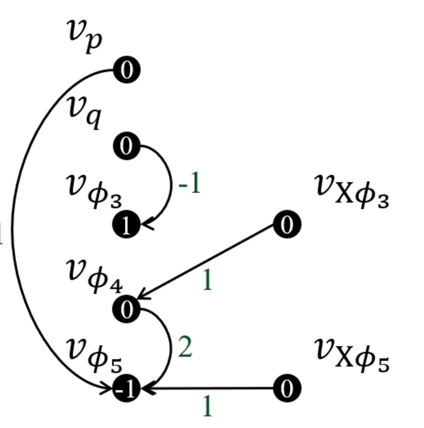
LTL$_f$ Graph

Definition 1 (simplified). *Let ϕ be an LTL$_f$ formula. Its* LTL$_f$ graph *G_ϕ is a four-tuple $(V_\phi, E_\phi, \mathsf{W}_\phi, \mathsf{b}_\phi)$ defined as follows,*

- *if* $\mathsf{unfold}(\phi_i) = p$, *then* $V_\phi = V_\phi \cup \{v_p\}$ *and* $\mathsf{b}_\phi(v_p) = 0$;
- *if* $\mathsf{unfold}(\phi_i) = \neg\phi_j$, *then* $V_\phi = V_\phi \cup \{v_{\phi_j}\}$, $E_\phi = E_\phi \cup \{\langle v_{\phi_j}, v_{\phi_i}\rangle\}$, $\mathsf{W}_\phi(\langle v_{\phi_j}, v_{\phi_i}\rangle) = -1$, *and* $\mathsf{b}_\phi(v_{\phi_i}) = 1$;
- *if* $\mathsf{unfold}(\phi_i) = \phi_j \wedge \phi_k$, *then* $V_\phi = V_\phi \cup \{v_{\phi_j}, v_{\phi_k}\}$, $E_\phi = E_\phi \cup \{\langle v_{\phi_j}, v_{\phi_i}\rangle, \langle v_{\phi_k}, v_{\phi_i}\rangle\}$, $\mathsf{W}_\phi(\langle v_{\phi_j}, v_{\phi_i}\rangle) = 1$, $\mathsf{W}_\phi(\langle v_{\phi_k}, v_{\phi_i}\rangle) = 1$, *and* $\mathsf{b}_\phi(v_{\phi_i}) = -1$;
- *if* $\mathsf{unfold}(\phi_i) = \mathsf{X}\phi_j$, *then* $V_\phi = V_\phi \cup \{v_{\mathsf{X}\phi_j}\}$, $E_\phi = E_\phi \cup \{\langle v_{\mathsf{X}\phi_j}, v_{\phi_i}\rangle\}$, $\mathsf{W}_\phi(\langle v_{\mathsf{X}\phi_j}, v_{\phi_i}\rangle) = 1$, *and* $\mathsf{b}_\phi(v_{\phi_i}) = \mathsf{b}_\phi(v_{\mathsf{X}\phi_j}) = 0$;
- *if* $\mathsf{unfold}(\phi_i) = \phi_k \vee (\phi_j \wedge \mathsf{X}\phi_i)$, *then* $V_\phi = V_\phi \cup \{v_{\phi_k}, v_{\phi_j}, v_{\mathsf{X}\phi_i}\}$, $E_\phi = E_\phi \cup \{\langle v_{\phi_k}, v_{\phi_i}\rangle, \langle v_{\phi_j}, v_{\phi_i}\rangle, \langle v_{\mathsf{X}\phi_i}, v_{\phi_i}\rangle\}$, $\mathsf{W}_\phi(\langle v_{\phi_k}, v_{\phi_i}\rangle) = 2$, $\mathsf{W}_\phi(\langle v_{\phi_j}, v_{\phi_i}\rangle) = 1$, $\mathsf{W}_\phi(\langle v_{\mathsf{X}\phi_i}, v_{\phi_i}\rangle) = 1$, $\mathsf{b}_\phi(v_{\phi_i}) = -1$, *and* $\mathsf{b}_\phi(v_{\mathsf{X}\phi_i}) = 0$,

where $p \in \mathbb{P} \cup \{\top, \bot\}$ and ϕ_j, ϕ_k are LTL$_f$ formulae.

Example 1. *Let $\phi = p\mathsf{U}(\mathsf{X}\neg q)$ be an LTL$_f$ formula.* $\mathsf{sub}(\phi) = \{p, q, \phi_3, \phi_4, \phi_5\}$, *where $\phi_3 = \neg q$, $\phi_4 = \mathsf{X}\phi_3$, and $\phi_5 = p\mathsf{U}\phi_4$.*



- $\phi_5 = \phi_4 \vee (p \wedge \mathsf{X}\phi_5)$
- *Relation of satisfaction of p, ϕ_4, $\mathsf{X}\phi_5$, and ϕ_5:*
  - *if p is $\bot$, ϕ_4 is $\bot$, and $\mathsf{X}\phi_5$ is $\bot$, then ϕ_5 is $\bot$;*
  - *if p is $\bot$, ϕ_4 is $\bot$, and $\mathsf{X}\phi_5$ is $\top$, then ϕ_5 is $\bot$;*
  - *if p is $\bot$, ϕ_4 is $\top$, and $\mathsf{X}\phi_5$ is $\bot$, then ϕ_5 is $\top$;*
  - ...
- **Boolean → real**: $\bot \to \leq 0$ *and* $\top \to \geq 1$

State Classifier

Definition 2 (simplified). *Let ϕ be an LTL$_f$ formula such that $|\mathsf{sub}(\phi)| = L$, $(V_\phi, E_\phi, \mathsf{W}_\phi, \mathsf{b}_\phi)$ an LTL$_f$ graph of ϕ, and $\pi = s_0, s_1, \ldots, s_n$ a trace. $(\mathbf{x}^{(0)}_{s_i})_j = 1$ if $\phi_j \in s_i$ or $(\mathbf{x}^{(0)}_{s_i})_j = 0$ otherwise. $\mathbf{x}^{(t)}_{s_i}$ is defined as follows:*

$$\mathbf{x}^{(t)}_{s_i} = \sigma(\mathbf{C}_\phi \mathbf{x}^{(t-1)}_{s_i} + \mathbf{A}_\phi \mathbf{x}_{s_{i+1}} + \mathbf{b}_\phi), \tag{1}$$

$$(\mathbf{C}_\phi)_{ij} = \begin{cases} \mathsf{W}_\phi(\langle v_{\phi_j}, v_{\phi_i}\rangle), & \text{if } \langle v_{\phi_j}, v_{\phi_i}\rangle \in E_\phi \text{ and } \phi_j \in \mathsf{sub}(\phi) \\ 0, & \text{otherwise,} \end{cases}$$

$$(\mathbf{A}_\phi)_{ij} = \begin{cases} \mathsf{W}_\phi(\langle v_{\phi_j}, v_{\phi_i}\rangle), & \text{if } \langle v_{\phi_j}, v_{\phi_i}\rangle \in E_\phi \text{ and } \phi_j \in \{\mathsf{X}\phi_k \mid \phi_k \in \mathsf{sub}(\phi)\} \\ 0, & \text{otherwise,} \end{cases}$$

$$(\mathbf{b}_\phi)_i = \mathsf{b}_\phi(v_{\phi_i}), \quad \text{for all } v_{\phi_i} \in V_\phi \text{ and } \phi_i \in \mathsf{sub}(\phi).$$

By $\mathcal{S}_\phi$ we denote the state classifier *$\mathcal{S}_\phi$, i.e., $\mathbf{x}^{(T)}_{s_i} = \mathcal{S}_\phi(\mathbf{x}^{(0)}_{s_i}, \mathbf{x}_{s_{i+1}}, T)$.*

Trace Classifier

Definition 3. *Let ϕ be an LTL$_f$ formula such that $|\mathsf{sub}(\phi)| = L$, and $\pi = s_0, s_1, \ldots, s_n$ a trace. By $\mathcal{T}_\phi$ we denote the* trace classifier *which takes a vector $\mathbf{x}^{(0)}_{s_i}$ and a number of iterations $T \in \mathbb{N}$ as input and $\mathbf{x}^{(T)}_{s_i}$ as output; i.e., $\mathbf{x}^{(T)}_{s_i} = \mathcal{T}_\phi(\mathbf{x}^{(0)}_{s_i}, T)$, where*

$$\mathcal{T}_\phi(\mathbf{x}^{(0)}_{s_i}, T) = \begin{cases} \mathcal{S}_\phi(\mathbf{x}^{(0)}_{s_i}, \mathcal{T}_\phi(\mathbf{x}^{(0)}_{s_{i+1}}, T), T), & 0 \leq i < n \\ \mathcal{S}_\phi(\mathbf{x}^{(0)}_{s_i}, \mathbf{0}, T), & i = n \end{cases} \tag{2}$$

Relationship between GNNs and LTL$_f$

Theorem 1. *Let ϕ be an LTL$_f$ formula such that $|\mathsf{sub}(\phi)| = L$. For every trace $\pi = s_0, s_1, \ldots, s_n$, $(\mathcal{T}_\phi(\mathbf{x}^{(0)}_{s_0}, L))_L = 1$ if and only if $\pi \models \phi$.*

GLTLf: GNN-based Learning Approach

Framework. The framework of `GLTLf` is summarized as follows.

1. First train a simple homogeneous AC-GNN model $\mathcal{S}$ to distinguish positive and negative traces.
2. Then interpret the parameters of $\mathcal{S}$ to obtain an LTL$_f$ formula ϕ_A.

Converting Traces to Graphs. Let $\pi = s_0, s_1, \ldots, s_n$ be a trace. Its *directed path graph* $G_\pi = (V_\pi, E_\pi)$:

- each vertex v_i in V_π corresponds to a state s_i in π;
- for each pair of states s_i, s_{i+1} in π, there is an edge $\langle v_{i+1}, v_i\rangle$ in E_π.

Training GNNs as Classifiers. Each vertex $v_i \in V_\pi$ has a feature vector $\mathbf{x}_{v_i}$ ($[x_1, \ldots, x_{|\mathbb{P}|}, x_\top, x_{|\mathbb{P}|+2}, \ldots, x_L]$).

- State classifier $\mathcal{S}$ follows Definition 2, but $\mathbf{C}$, $\mathbf{A}$, and $\mathbf{b}$ need to be trained.
- Trace classifier $\mathcal{T}$ is defined by Definition 3.
  - $(\mathcal{T}(\mathbf{x}^{(0)}_{v_i}, L))_j$ indicates whether $\pi_i \models \phi_j$, where $\phi_j \in \mathsf{sub}(\phi)$.

GLTLf: GNN-based Learning Approach

Interpreting. Interpret a sub-formula from $(\mathbf{C})_i$, $(\mathbf{A})_i$, and $(\mathbf{b})_i$.

1. First recommend a set of candidate formulae based on the *interpretation similarity*.
2. Then select the best formula based on the *discrimination effect for the traces*.

Result Analysis

Competitor.

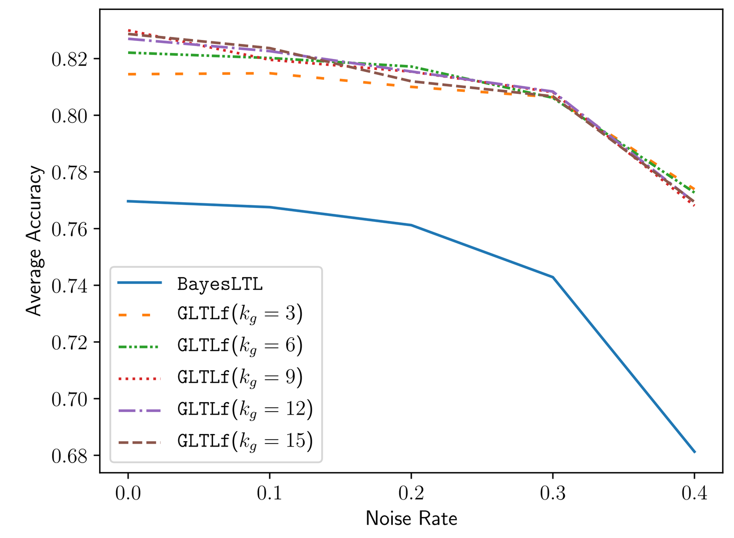
approach	noisy data	arbitrary formulae
`C.&M.` (Camacho and McIlraith 2019)	×	✓
`BayesLTL` (Kim et al. 2019)	✓	×
`MaxSAT-DT` (Gaglione et al. 2021)	✓	✓
`GLTLf` (Ours)	✓	✓

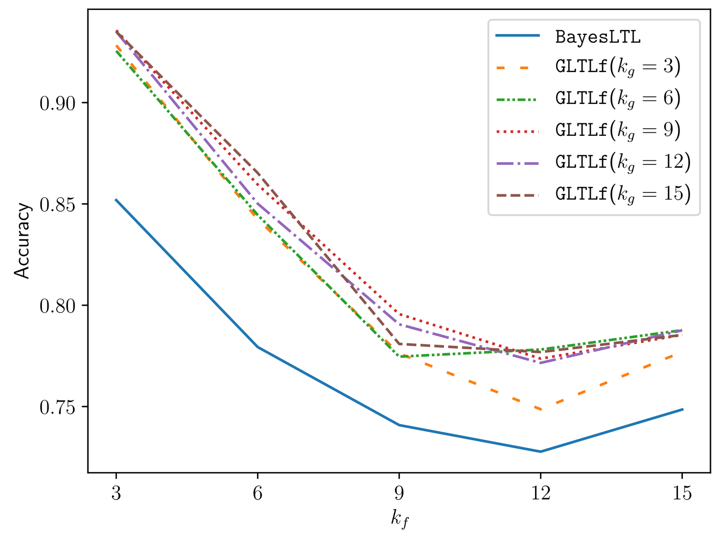
Comparisons on Noise-free Data.

	$k_f = 3$			$k_f = 6$			$k_f = 9$			$k_f = 12$			$k_f = 15$		
	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`	93.50	93.09	50	86.53	86.28	50	78.09	78.66	50	77.69	78.63	50	78.53	79.34	50

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

Comparisons on Noise Data, Scalability, and Robustness.





- `GLTLf` is proven to be more noise-tolerated then other approaches.
- `GLTLf` can handle long formulae with higher performance compared to other approaches.
- `GLTLf` is able to solve formulae in various sizes and is rather robust.
  - Larger networks achieve better accuracies, even when solving short formulae.
  - Small networks can also get good performance on long formulae.

Acknowledgments

We thank Rongzhen Ye for discussion on the paper and anonymous referees for helpful comments. This paper was supported by the National Natural Science Foundation of China (No. 61876204 and No. 61976232), Humanities and Social Science Research Project of Ministry of Education (18YJCZH006).