

# TECHS: Temporal Logical Graph Networks for Explainable Extrapolation Reasoning

Qika Lin<sup>1,2</sup>, Jun Liu<sup>1,3\*</sup>, Rui Mao<sup>4</sup>, Fangzhi Xu<sup>1,2</sup>, Erik Cambria<sup>4</sup>

<sup>1</sup>School of Computer Science and Technology, Xi'an Jiaotong University

<sup>2</sup>Shaanxi Provincial Key Laboratory of Big Data Knowledge Engineering

<sup>3</sup>National Engineering Lab for Big Data Analytics

<sup>4</sup>School of Computer Science and Engineering, Nanyang Technological University

qikalin@foxmail.com, liukeen@xjtu.edu.cn, rui.mao@ntu.edu.sg,

Leo981106@stu.xjtu.edu.cn, cambria@ntu.edu.sg

## Abstract

Extrapolation reasoning on temporal knowledge graphs (TKGs) aims to forecast future facts based on past counterparts. There are two main challenges: (1) incorporating the complex information, including structural dependencies, temporal dynamics, and hidden logical rules; (2) implementing differentiable logical rule learning and reasoning for explainability. To this end, we propose an explainable extrapolation reasoning framework **T**emporal **logi****C**al **grap****H** **n**e**t**w**o****r****k**s (TECHS), which mainly contains a temporal graph encoder and a logical decoder. The former employs a graph convolutional network with temporal encoding and heterogeneous attention to embed topological structures and temporal dynamics. The latter integrates propositional reasoning and first-order reasoning by introducing a reasoning graph that iteratively expands to find the answer. A forward message-passing mechanism is also proposed to update node representations, and their propositional and first-order attention scores. Experimental results demonstrate that it outperforms state-of-the-art baselines.

## 1 Introduction

Knowledge Graphs (KGs) are widely used in intelligent systems (Ji et al., 2022; Mao et al., 2022; Zhu et al., 2023), where knowledge is commonly represented by triplets in the form of  $(s, r, o)$ . The limit of conventional KGs is that real-world knowledge usually evolves over time. For example, a fact (*Donald Trump, presidentOf, USA*) is incorrect now because *Joe Biden* has been the new president of the USA since 2021. For more comprehensive representations of knowledge, Temporal Knowledge Graphs (TKGs) (Liang et al., 2022) are proposed by introducing time information (time point or interval) via quadruplets, i.e.,  $(s, r, o, t)$ . Then, the former example is defined as (*Donald Trump, presidentOf, USA, 2017/01/20-2021/01/20*).

\* Corresponding author.

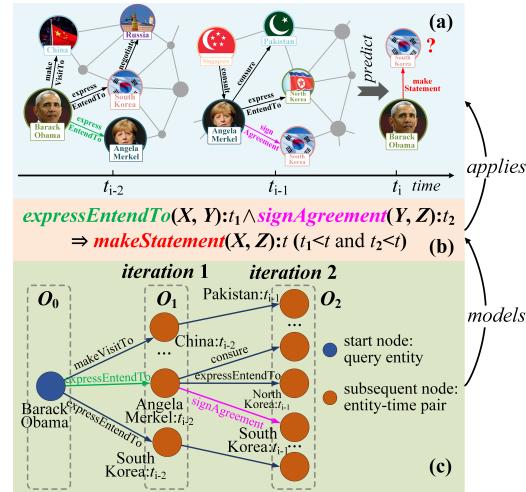


Figure 1: (a) Illustration of a TKG and extrapolation reasoning. (b) An example of temporal rules that can be applied to answer the query in (a). (c) An example of the reasoning graph that is capable of modeling rule (b).

TKGs are usually incomplete (Cai et al., 2022; Liang et al., 2022). Many studies predicted future facts, based on past facts, namely TKG forecasting or extrapolation reasoning. Figure 1a shows the task that predicts facts at time  $t_i$  with the facts at  $t_{i-2}$  and  $t_{i-1}$ . A model should not only learn topology dependencies, i.e., the neighbor information of an entity (like *Barack Obama* at  $t_{i-2}$ ), but also learn temporal dynamics, i.e., the variations of properties of an entity over time (e.g., *Angela Merkel* evolves during  $t_{i-2}$  to  $t_{i-1}$ ). Thus, temporal embedding methods, e.g., TNT-ComplEx (Lacroix et al., 2020) and CyGNet (Zhu et al., 2021) were proposed. However, these black-box methods fail to explain their predictions. An explainable method, xERTE (Han et al., 2021) conducted instanced propositional reasoning. However, the model is not scalable, as the evidence is entity-dependent, e.g., related to *Barack Obama* and other entities in Figure 1a. If we can learn the entity-independent rule in Figure 1b for the query

$(Barack\ Obama, makeStatement, ?, t_i)$  in Figure 1a, the correct answer *South Korea* will be easily obtained after rule grounding.

Motivated by the fact that TKGs have many hidden logical rules to achieve explainable and accurate predictions, TLogic (Liu et al., 2022) searched first-order logical rules and used them for reasoning. However, this two-step pipeline method may cause error propagation issues. Generally, there are two main challenges for explainable extrapolation reasoning on TKGs: (1) TKGs contain diverse information, e.g., structural dependencies, temporal dynamics, and hidden logical rules that are difficult to incorporate together and achieve full coverage; (2) Logical rule representations are discrete and symbolic, resulting in the natural gap between logical rules and the continuous computation of neural networks. Thus, implementing differentiable logical rule learning and reasoning is not directly achievable (Yang et al., 2017).

To address above issues, we propose a unified framework **T**Emporal logi**C**al grap**H** network**S** (TECHS). It first utilizes a graph convolutional network (GCN) to embed topological structures and temporal dynamics. To determine the weights of different edges between entities, a generic time encoding and a heterogeneous attention mechanism is introduced. Then, a logical decoder is proposed to integrate propositional and first-order reasoning to find the answer. A reasoning graph that contains both query entity and entity-time pair nodes is used to constantly expand over iterations. We update propositional and first-order attention weights as well as node representations via a novel forward message-passing mechanism. Finally, nodes' attention weights with the same entity are aggregated as the answer indicator. Besides, first-order logical rules can be induced by a novel **F**orward **A**ttentive **R**ule **I**nduction (FARI) algorithm using learned first-order attention weights.

Our contributions are summarized as follows: (1) A unified framework TECHS is proposed to conduct explainable extrapolation reasoning on TKGs. To our best knowledge, this is the first study to jointly model structural dependencies, temporal dynamics, and propositional and first-order reasoning. (2) We integrate propositional and first-order reasoning in a logical decoder, where a forward message-passing is proposed to update their attention weights and node representations to achieve explainability. First-order logical rules are induced

by a novel FARI algorithm. (3) Extensive experiments verify the effectiveness of each module and the superiority over state-of-the-art baselines.

## 2 Related Work

The studies of extrapolation reasoning can be categorized into the following three trends.

**Static Embedding.** By omitting time information in fact quadruplets, general KG embedding methods can be utilized for TKGs, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016). However, these methods simply consider the structural dependency in TKGs and ignore the temporal dynamics.

**Temporal Embedding.** TTransE (Leblay and Chekol, 2018) expanded TransE to the temporal setting by fusing temporal information in relation embeddings. Similarly, TA-DistMult and TA-TransE (García-Durán et al., 2018) learned relation representations with time information and calculated quadruplet plausibility by DistMult and TransE. Differently, DE-SimplE (Goel et al., 2020) proposed diachronic entity embedding which contained static segment and time-varying segment. Upon ComplEx, TNTComplEx (Lacroix et al., 2020) learned complex-valued embeddings for the entity, relation and time. RE-Net (Jin et al., 2020) learned the global representations of the time subgraph and the local representations of nodes on it. CyGNet (Zhu et al., 2021) introduced a time-aware copy-generation mechanism to model the probability of existing facts, occurring in the future and predicted whether new facts would emerge. However, the aforementioned methods are all in black-box fashion and lack of explainability.

**Explainable Reasoning.** xERTE (Han et al., 2021) proposed a human-understandable reasoning strategy, introducing an expanding query-relevant subgraph to achieve explainability. TiTer (Sun et al., 2021) conducted reasoning from a query node and sequentially transferred to a new node related to the prior on TKGs until the answer was founded. Upon AnyBURL (Meilicke et al., 2019) that sampled paths to learn first-order rules in static KGs, TLogic (Liu et al., 2022) learned temporal logical rules with confidences via a temporal random walk. The candidate scores were obtained by rule applications in TKGs. However, xERTE and TiTer conducted propositional reasoning by an end-to-end framework that had limited scalability, as its reasoning process was query-specific. Although

TLogic learned query-independent first-order logical rules, its pipeline method might cause error propagation and performance degradation.

### 3 Preliminaries

A TKG can be represented as  $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}\}$ , where  $\mathcal{E}$ ,  $\mathcal{R}$  and  $\mathcal{T}$  denote the set of entity, relation and time, respectively.  $\mathcal{F} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$  is the fact collection. Each fact is a quadruplet, such as  $(s, r, o, t)$  where  $s, o \in \mathcal{E}$ ,  $r \in \mathcal{R}$  and  $t \in \mathcal{T}$ . For a query  $(\tilde{s}, \tilde{r}, ?, \tilde{t})$  in testing, the model needs to predict an answer entity  $\tilde{o}$ , based on the facts that occur earlier than  $\tilde{t}$ , i.e.,  $\tilde{t} > \max(\mathcal{T}_{train})$ .

Logical reasoning in KGs can be categorized as: propositional and first-order. Propositional reasoning, generally known as multi-hop reasoning (Ren and Leskovec, 2020; Zhang et al., 2021, 2022a), is entity-dependent that usually reasons over query-related paths to obtain an answer. First-order reasoning is entity-independent, using first-order logical (FOL) rules for different entities (Zhang et al., 2022b), describing causal knowledge in the form of *body* to *head*, e.g., *premise*  $\Rightarrow$  *conclusion*, where new facts can be deduced, given observed ones. For efficient and explainable reasoning on TKGs, we define the FOTH rule and the reasoning graph.

**Definition 1. First-order Temporal Horn (FOTH) Rule:** Based on Horn rules (Lin et al., 2022) on static KGs, atoms in FOTH rule body are connected transitively by shared variables. Meanwhile, rule body and rule head have the same start and end variables. Time growth also needs to be satisfied, i.e., time sequence is increasing and the time in the rule head is the maximum. For example, the following rule  $\epsilon$ ,  $\exists X, Y, Z \ r_1(X, Y) : t_1 \wedge r_2(Y, Z) : t_2 \Rightarrow r(X, Z) : t$  is a FOTH rule with length 2 if  $t_1 \leq t_2 < t$ .  $X$ ,  $Y$  and  $Z$  are variables that can be instantiated as entities of TKGs by rule grounding. Noticeably, for rule learning and reasoning,  $t_1$ ,  $t_2$  and  $t$  are virtual time variables that are only used to satisfy the time growth and do not have to be instantiated. To represent the rule certainty, each rule is assigned with a confidence value  $\epsilon \in [0, 1]$ .

**Definition 2. Reasoning Graph:** For a query  $(\tilde{s}, \tilde{r}, ?, \tilde{t})$ , we introduce a reasoning graph  $\tilde{\mathcal{G}} = \{\mathcal{O}, \mathcal{R}, \mathcal{F}\}$  for propositional and first-order reasoning.  $\mathcal{O}$  is a node set that consists of nodes in different iteration steps, i.e.,  $\mathcal{O} = \mathcal{O}_0 \cup \mathcal{O}_1 \cup \dots \cup \mathcal{O}_L$ .  $\mathcal{O}_0$  only contains a query entity  $\tilde{s}$  and others consist of nodes in the form of entity-time pairs.  $(n_i^l, \bar{r}, n_j^{l+1}) \in \tilde{\mathcal{F}}$  is an edge that links nodes at

two neighbor steps, i.e.,  $n_i^l \in \mathcal{O}_l$ ,  $n_j^{l+1} \in \mathcal{O}_{l+1}$  and  $\bar{r} \in \mathcal{R}$ . The reasoning graph is constantly expanded by searching for posterior neighbor nodes. For start node  $n^0 = \tilde{s}$ , its posterior neighbors are  $\mathcal{N}(n^0) = \{(e_i, t_i) | (\tilde{s}, \bar{r}, e_i, t_i) \in \mathcal{F} \wedge t_i < \tilde{t}\}$ . For a node in following steps  $n_i^l = (e_i, t_i) \in \mathcal{O}_l$ , its posterior neighbors are  $\mathcal{N}(n_i^l) = \{(e_j, t_j) | (e_i, \bar{r}, e_j, t_j) \in \mathcal{F} \wedge t_i \leq t_j \wedge t_j < \tilde{t}\}$ . Its prior parents are  $\mathcal{N}(n_i^l) = \{(n_j^{l-1}, \bar{r}) | n_j^{l-1} \in \mathcal{O}_{l-1} \wedge (n_j^{l-1}, \bar{r}, n_i^l) \in \tilde{\mathcal{F}}\}$ . An example reasoning graph with two steps is shown in Figure 1c. To take prior nodes into account at the current step, an extra relation *self* is added. Then,  $n_i^l = (e_i, t_i)$  can be obtained at the next step as  $n_i^{l+1} = (e_i, t_i)$  ( $t_i$  is the minimum time if  $l = 0$ ).

### 4 Methodology

There are three key technical parts in TECHS: temporal graph encoder, logical decoder, and extrapolation prediction. Figure 2 shows its architecture.

#### 4.1 Temporal Graph Encoder

Generally, GCNs follow an iterative message-passing strategy to continuously aggregate information from neighbor nodes. As conventional GCNs cannot model time information, we propose a temporal graph encoder. The generic time encoding (Xu et al., 2020) is introduced to embed times in TKGs as it is fully compatible with attention to capture temporal dynamics, which is defined as:  $\mathbf{e}_t = \sqrt{\frac{1}{d_t}} [\cos(w_1 t + b_1), \dots, \cos(w_{d_t} t + b_{d_t})]$ .  $[w_1, \dots, w_{d_t}]$  and  $[b_1, \dots, b_{d_t}]$  are trainable parameters for transformation weights and biases.  $d_t$  is the dimension of time embedding. Based on it, a temporal GCN is proposed by fusing neighbor information with the heterogeneous attention:

$$\mathbf{h}_o^{k+1} = \mathbf{W}_{h1}^k \mathbf{h}_o^k + \sum_{(s,r,t) \in \hat{\mathcal{N}}(o)} \alpha_{s,r,o,t}^k \mathbf{W}_{h2}^k \mathbf{m}_{s,r,t}^k, \quad (1)$$

where  $\mathbf{W}$  denotes a transformation matrix.  $\hat{\mathcal{N}}$  is the neighbor set.  $\mathbf{m}_{s,r,t}^k$  is the message information of neighbors that contains subject, relation and time representations, which is given by:

$$\mathbf{m}_{s,r,t}^k = \mathbf{W}_{m1}^k [(\mathbf{h}_s^k + \mathbf{e}_t) \odot (\mathbf{g}_r^k + \mathbf{e}_t)]. \quad (2)$$

$\mathbf{h}$  and  $\mathbf{g}$  are the entity and relation embeddings, respectively.  $\odot$  is the element-wise product of two embedding vectors.  $\alpha_{s,r,o,t}^k$  is a heterogeneous attention value to determine the importance of a current temporal edge. It is obtained by the correlation

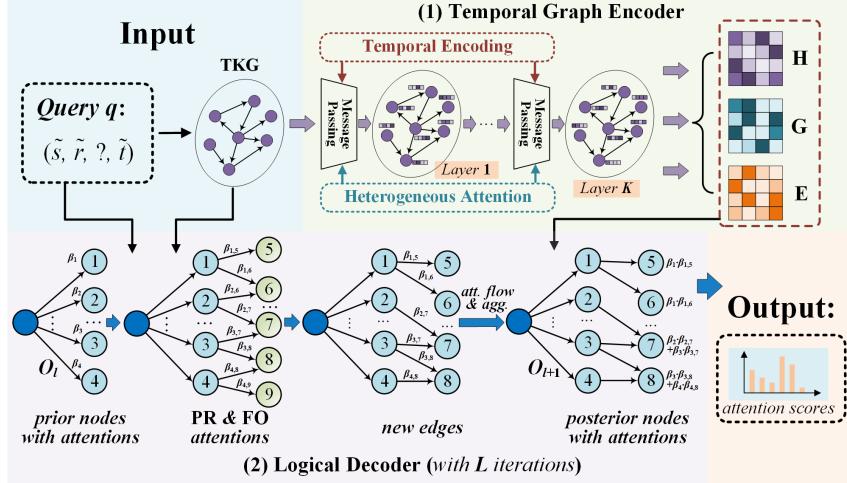


Figure 2: An overview of the TECHS. The temporal graph encoder utilizes temporal encoding and heterogeneous attention for structural dependencies and temporal dynamics. The logical decoder combines propositional (PR) and first-order (FO) reasoning by continuously conducting forward message-passing in the reasoning graph.

between time, relation and the current entities:

$$a_{s,r,o,t}^k = \sigma((\alpha^k)^\top \mathbf{W}_a [\mathbf{e}_t \| \mathbf{g}_r^k \| (\mathbf{h}_s^k - \mathbf{h}_o^k)]),$$

$$\alpha_{s,r,o,t}^k = \frac{\exp(a_{s,r,o,t}^k)}{\sum_{(s',r',t') \in \hat{\mathcal{N}}(o)} \exp(a_{s',r',o,t'}^k)}, \quad (3)$$

where  $\sigma$  is *LeakyReLU* (Xu et al., 2015).  $\|$  is concatenation.  $\alpha^k$  is the attention vector to be learned.

Finally, the relation embedding is updated by  $\mathbf{g}_r^{k+1} = \mathbf{W}_r^k \mathbf{g}_r^k$ . At the last layer  $K$ , the representation matrix  $\mathbf{H}$ ,  $\mathbf{G}$  and  $\mathbf{E}$  of entity, relation and time are obtained, then feeding into the logical decoder.

## 4.2 Logical Decoder

For decoding the answer for query  $(\tilde{s}, \tilde{r}, ?, \tilde{t})$ , we introduce an iterative forward message-passing mechanism in a continuously expanding reasoning graph, regulated by propositional and first-order reasoning. In the reasoning graph, we set three learnable parameters for each node  $n_i^l$  to guide the computation: node embedding  $\mathbf{n}_i^l$ , hidden FOTH embedding  $\mathbf{o}_{n_i^l}$  and reasoning attention  $\beta_{n_i^l}$ . The start node  $n^0 = \tilde{s}$  is initialized as its embedding  $\mathbf{h}_{\tilde{s}}$ . A hidden FOTH representation  $\mathbf{o}_{n^0}$  for  $n^0$  is initialized as a query relation embedding  $\mathbf{g}_{\tilde{r}}$ . The attention weight  $\beta_{n^0}$  for  $n^0$  is initialized as 1. The node  $n_i = (e_i, t_i)$  are firstly represented by the linear transformation of GCN embeddings:  $\mathbf{n}_i = \mathbf{W}_n [\mathbf{h}_{e_i} \| \mathbf{e}_{t_i}]$ . Constant forward computation is required in the reasoning sequence of the target, whether conducting multi-hop propositional reasoning or first-order logic reasoning. Thus, forward message-passing is proposed to pass information (i.e., representations and attention weights) from the prior nodes to their posterior

neighbor nodes. The computation of each node is contextualized with prior information that contains both entity-dependent and entity-independent parts, reflecting the continuous accumulation of knowledge and credibility in the reasoning process.

Specifically, to update node embeddings in step  $l+1$ , its own feature and the information from its priors are integrated:

$$\mathbf{n}_j^{l+1} = \mathbf{W}_{n1}^l \mathbf{n}_j + \sum_{(n_i^l, \bar{r}) \in \hat{\mathcal{N}}(n_j^{l+1})} \beta_{n_i^l, \bar{r}, n_j^{l+1}} \mathbf{W}_{n2}^l \mathbf{m}_{n_i^l, \bar{r}, n_j^{l+1}}, \quad (4)$$

where  $\mathbf{m}_{n_i^l, \bar{r}, n_j^{l+1}}$  is the message from a prior node to its posterior node, which is given by the node and relation representations:

$$\mathbf{m}_{n_i^l, \bar{r}, n_j^{l+1}} = \mathbf{W}_{m2}^l [\mathbf{n}_i^l \| \mathbf{g}_{\bar{r}} \| \mathbf{n}_j]. \quad (5)$$

This updating form superficially seems similar to the general message-passing in GCNs. However, they are actually different as ours is in a one-way and hierarchical manner, which is tailored for the tree-like structure of the reasoning graph.

The attention weight  $\beta_{n_i^l, \bar{r}, n_j^{l+1}}$  for each edge in a reasoning graph contains two parts: propositional and first-order attention. As propositional attention is entity-dependent, we compute it by the semantic association of entity-dependent embeddings between the message and the query:

$$e_{n_i^l, \bar{r}, n_j^{l+1}}^1 = \text{SIGMOID}(\mathbf{W}_p^l [\mathbf{m}_{n_i^l, \bar{r}, n_j^{l+1}} \| \mathbf{q}]), \quad (6)$$

where  $\mathbf{q} = \mathbf{W}_q [\mathbf{h}_{\tilde{s}} \| \mathbf{g}_{\tilde{r}} \| \mathbf{e}_{\tilde{t}}]$  is the query embedding. As first-order reasoning focuses on the interaction

among entity-independent relations, we first obtain the hidden FOTH embedding of an edge by fusing the hidden FOTH embedding of the prior node and current relation representation via a gated recurrent unit (GRU) (Chung et al., 2014). Then, the first-order attention is given by:

$$\begin{aligned}\mathbf{o}_{n_i^l, \bar{r}, n_j^{l+1}} &= \text{GRU}(\mathbf{g}_{\bar{r}}, \mathbf{o}_{n_i^l}), \\ e_{n_i^l, \bar{r}, n_j^{l+1}}^2 &= \text{SIGMOID}(\mathbf{W}_f^l \mathbf{o}_{n_i^l, \bar{r}, n_j^{l+1}}).\end{aligned}\quad (7)$$

Furthermore, the overall reasoning attention can be obtained by incorporating propositional and first-order parts to realize the complementarity of these two reasoning methods. Since the prior node with high credibility leads to faithful subsequent nodes, the attention of the prior flows to the current edge. Then, the softmax normalization is utilized to scale edge attentions on this iteration to [0,1]:

$$\begin{aligned}e_{n_i^l, \bar{r}, n_j^{l+1}} &= \beta_{n_i^l}(e_{n_i^l, \bar{r}, n_j^{l+1}}^1 + \lambda e_{n_i^l, \bar{r}, n_j^{l+1}}^2), \\ \beta_{n_i^l, \bar{r}, n_j^{l+1}} &= \frac{\exp(e_{n_i^l, \bar{r}, n_j^{l+1}}^1)}{\sum_{(n_i^l, \bar{r}') \in \tilde{\mathcal{N}}(n_j^{l+1})} \exp(e_{n_i^l, \bar{r}', n_j^{l+1}})},\end{aligned}\quad (8)$$

where  $\lambda$  is the weight for balancing the two reasoning types. Finally, the FOTH representation and attention of a new node  $n_j^{l+1}$  are aggregated from edges for the next iteration:

$$\begin{aligned}\mathbf{o}_{n_j^{l+1}} &= \sum_{(n_i^l, \bar{r}) \in \tilde{\mathcal{N}}(n_j^{l+1})} \beta_{n_i^l, \bar{r}, n_j^{l+1}} \mathbf{o}_{n_i^l, \bar{r}, n_j^{l+1}}, \\ \beta_{n_j^{l+1}} &= \sum_{(n_i^l, \bar{r}) \in \tilde{\mathcal{N}}(n_j^{l+1})} \beta_{n_i^l, \bar{r}, n_j^{l+1}}.\end{aligned}\quad (9)$$

**Insights of FOTH Rule Learning and Reasoning.** In general, the learning and reasoning of first-order logical rules on KGs or TKGs are usually in two-step fashion (Galárraga et al., 2013, 2015; Qu and Tang, 2019; Zhang et al., 2019; Qu et al., 2021; Vardhan et al., 2020; Liu et al., 2022; Cheng et al., 2022; Lin et al., 2023). First, it searches over whole data to mine rules and their confidences. Second, for a query, the model instantiates all variables to find all groundings of learned rules and then aggregates all confidences of eligible rules. For example, for a target entity  $o$ , its score can be the sum of learned rules with valid groundings and rule confidences can be modeled by a GRU. However, this is apparently not differentiable and cannot be optimized by an end-to-end manner. Thus, our model conducts the transformation of merging multiple

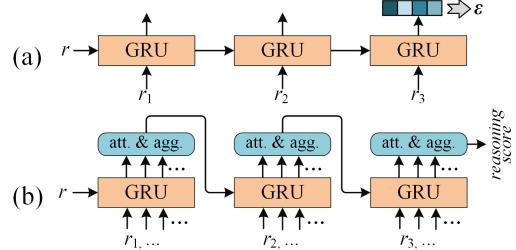


Figure 3: Illustration of insights of FOTH rule learning and reasoning. (a) Learning rule confidence for a specific rule  $r_1(X, Y_1) \wedge r_2(Y_1, Y_2) \wedge r_3(Y_2, Z) \Rightarrow r(X, Z)$  (time information is omitted). (b) Rule learning and reasoning process in TECHS, which performs attention aggregation of possible relations at each step to realize differentiable computing.

rules by merging possible relations at each step, using first-order attention as:

$$\begin{aligned}S_o &= \sum_{\gamma \in \Gamma} \beta_\gamma \\ &= \sum_{\gamma \in \Gamma} f[\text{GRU}(\mathbf{g}_{\gamma,h}, \mathbf{g}_{\gamma,b^1}, \dots, \mathbf{g}_{\gamma,b^{|\gamma|}})] \\ &\approx \prod_{l=1}^L \sum_{n_j \in \mathcal{O}_l} \bar{f}_l[\text{GRU}(\mathbf{g}_{\bar{r}}, \mathbf{o}_{n_j}^l)].\end{aligned}\quad (10)$$

$\beta_\gamma$  is the confidence of rule  $\gamma$ .  $\mathbf{g}_{\gamma,h}$  and  $\mathbf{g}_{\gamma,b^i}$  are the relation embeddings of head  $h$  and  $i$ -th body  $b^i$  of this rule.  $\bar{f}_l$  is for the attention calculation. In this way, the differentiable process is achieved. This is an extension and progression of Neural-LP (Yang et al., 2017) and DURM (Sadeghian et al., 2019) on TKGs. Figure 3 intuitively illustrates such transformation. Finally, the real FOTH rules can be easily induced to constantly perform attention calculation over the reasoning graph, which is summarized as FARI in Algorithm 1.

### 4.3 Extrapolation Prediction

After attention weights for nodes in the last decoding step  $L$  have been obtained, we can aggregate node attentions with the same entity to get the entity score:  $S_o = \sum_{n_i^L=(o, t_i)} \beta_{n_i^L}$ . All entity scores can be normalized into [0,1] by  $\hat{y}_o = \frac{S_o}{\sum_p S_p}$ . Compared with the true label  $y_o$ , the model can be optimized by a binary cross-entropy loss:

$$\mathcal{L} = - \sum_o y_o \log(\hat{y}_o) + (1 - y_o)(1 - \log(\hat{y}_o)). \quad (11)$$

The number of nodes may explode in the logical decoder as it shows an exponential increase to

---

**Algorithm 1:** FARI for FOTH rules.

---

**Input:** the reasoning graph  $\tilde{\mathcal{G}}$ , attentions  $e^2$ .  
**Output:** the FOTH rule set  $\Gamma$ .

```

1 Init  $\Gamma = \emptyset$ ,  $B(n_{\tilde{s}}^0) = [0, []]$ ,  $\mathcal{D}_0[n_{\tilde{s}}^0] = [1, B(n_{\tilde{s}}^0)]$ ;  

2 for  $l=1$  to  $L$  of decoder iterations do  

3   Initialize node-rule dictionary  $\mathcal{D}_l$ ;  

4   for node  $n_j^l$  in  $\mathcal{O}_l$  do  

5     Set rule body list  $B(n_j^l) = []$ ;  

6     for  $(n_i^{l-1}, \tilde{r})$  of  $\tilde{\mathcal{N}}(n_j^l)$  in  $\mathcal{O}_{l-1}$  do  

7       Prior  $e_{i,l-1}^2, B(n_i^{l-1}) = \mathcal{D}_{l-1}[n_i^{l-1}]$ ;  

8       for weight  $\epsilon$ , body  $\gamma_b$  in  $B(n_i^{l-1})$  do  

9          $\epsilon' = e_{i,l-1}^2 \cdot e_{n_i^{l-1}, \tilde{r}, n_j^l}^2$ ;  

10         $\gamma'_b = \gamma_b.add(\tilde{r})$ ,  

11         $B(n_j^l).add([\epsilon', \gamma'_b])$ ;  

12      Add  $n_j^l: [e_{j,l}^2, B(n_j^l)]$  to  $\mathcal{D}_l$ ;  

13    Normalize  $e_{j,l}^2$  of  $n_j^l$  in  $\mathcal{O}_l$  using softmax;  

14  for  $n_i^L$  in  $\mathcal{O}_L$  do  

15     $e_{i,L}^2, B(n_i^L) = \mathcal{D}_L[n_i^L]$ ;  

16    for  $\epsilon, \gamma_b$  in  $B(n_i^L)$  do  

17       $\Gamma.add([\epsilon, \gamma_b[1](X, Y_1) : t_1 \wedge \dots \wedge$   

        $\gamma_b[L](Y_{L-1}, Z) : t_L \Rightarrow \tilde{r}(X, Z) : t])$   

18 Return rule set  $\Gamma$ .

```

---

reach  $|\mathcal{N}(n_i)|^L$  by iterations. For computational efficiency, posterior neighbors of each node are sampled with a maximum of  $M$  nodes in each iteration. For sampling  $M$  node in the reasoning graph, we follow a time-aware weighted sampling strategy, considering that recent events may have a greater impact on the forecast target. Specifically, for a posterior neighbor node with time  $t'$ , we compute its sampling weight by  $\frac{\exp(t' - \bar{t})}{\sum_{\tilde{t}} \exp(\tilde{t} - \bar{t})}$  for the query  $(\tilde{s}, \tilde{r}, ?, \tilde{t})$ , where  $\bar{t}$  denotes the time of all possible posterior neighbor nodes for a prior node. After computing attention weights for each edge in the same iteration, we select top- $N$  among them with larger attention weights and prune others. As we add an extra *self* relation in the reasoning graph, the FARI algorithm can obtain all possible rules (no longer than length  $L$ ) by deleting existing atoms with the *self* relation in induced FOTH rules.

## 5 Experiments and Results

### 5.1 Datasets and Experiment Setup

We conduct experiments on five common TKG datasets for extrapolation reasoning, i.e., ICEWS14, ICEWS18, ICEWS0515, WIKI (Leblay and Chekol, 2018) and YAGO (Mahdisoltani et al., 2015), which are the union ones of model xERTE, TITer and TLogic. The first three are all the

Dataset	$ \mathcal{E} $	$ \mathcal{R} $	$ \mathcal{T} $	$ \mathcal{F}_{train} $	$ \mathcal{F}_{valid} $	$ \mathcal{F}_{test} $
ICEWS14	7,128	230	365	63,685	13,823	13,222
ICEWS18	23,033	256	304	373,018	45,995	49,545
ICEWS0515	10,488	251	4,017	322,958	69,224	69,147
WIKI	12,554	24	232	539,286	67,538	63,110
YAGO	10,623	10	189	16,1540	19,523	20,026

Table 1: The statistics of five TKG datasets.

subsets of Integrated Crisis Early Warning System (O’brien, 2010). The last two contain massive real facts that are distinguished by years. The statistics of these five datasets are detailed in Table 1.

For training and testing, we add an inverse relation for each relation in TKGs. Thus, for the head entity prediction of query  $(?, \tilde{r}, \tilde{o}, \tilde{t})$ , we can predict results by its variant  $(\tilde{o}, \tilde{r}^{-1}, ?, \tilde{t})$ . For testing, *time-filter* setting is used in which all correct entities at the query time except for the true query object are filtered out from answers. For entities out of the final iteration of the reasoning graph, we set their scores as 0. Mean reciprocal rank (MRR) and Hits@ $k$  (H@ $k$  for abbreviation,  $k$  is 1, 3 or 10) are selected as evaluation metrics, where larger values denote better performance. The above settings are all in line with baselines for equal comparison.

We introduce fourteen baselines in three technical trends: (1) **Static Embedding**: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016). (2) **Temporal Embedding**: TTransE (Leblay and Chekol, 2018), TA-DistMult (García-Durán et al., 2018), TA-TransE (García-Durán et al., 2018), DE-SimplE (Goel et al., 2020), TNTComplEx (Lacroix et al., 2020), RE-Net (Jin et al., 2020) and CyGNet (Zhu et al., 2021). (3) **Explainable Reasoning**: xERTE (Han et al., 2021), TITer (Sun et al., 2021), AnyBURL (Meilicke et al., 2019) and TLogic (Liu et al., 2022). When conducting experiments, the default max number of sampled nodes and selected edges are 600 and 100, respectively. The learning rate, GCN layers, GCN dimensions, iteration steps, decoder dimensions and first-order weight  $\lambda$  are set to 0.001, 2, 200, 3, 50 and 0.65 by default. Adam algorithm (Kingma and Ba, 2015) is utilized to optimize the model parameters. When conducting experiments, our model is implemented in DGL (Wang et al., 2019) and PyTorch (Paszke et al., 2019), and trained on a single GPU of NVIDIA Tesla V100 with 32G memory.

### 5.2 Comparison Results

In each dataset, we run five times with different random seeds and report their mean results in Ta-

Model	ICEWS14				ICEWS0515				ICEWS18			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	22.48	13.36	25.63	41.23	22.55	13.05	25.61	42.05	12.24	5.84	12.81	25.10
DistMult	27.67	18.16	31.15	46.96	28.73	19.33	32.19	47.54	10.17	4.52	10.33	21.25
ComplEx	30.84	21.51	34.48	49.58	31.69	21.44	35.74	52.04	21.01	11.87	23.47	39.87
TTransE	13.43	3.11	17.32	34.55	15.71	5.00	19.72	38.02	8.31	1.92	8.56	21.89
TA-DistMult	26.47	17.09	30.22	45.41	24.31	14.58	27.92	44.21	16.75	8.61	18.41	33.59
TA-TransE	17.41	0.00	29.19	47.41	19.37	1.81	31.34	50.33	12.59	0.01	17.92	37.38
DE-SimplE	32.67	24.43	35.69	49.11	35.02	25.91	38.99	52.75	19.30	11.53	21.86	34.80
TNTComplEx	32.12	23.35	36.03	49.13	27.54	19.52	30.80	42.86	21.23	13.28	24.02	36.91
RE-Net	38.28	28.68	41.34	54.52	42.97	31.26	46.85	63.47	28.81	19.05	32.44	47.51
CyGNet	32.73	23.69	36.31	50.67	34.97	25.67	39.09	52.94	24.93	15.90	28.28	42.61
xERTE <sup>†</sup>	40.79	32.70	45.67	57.30	46.62	<u>37.84</u>	52.31	63.92	29.31	21.03	33.51	46.48
TITer <sup>†</sup>	41.73	32.74	46.46	58.44	—	—	—	—	<u>29.98</u>	<b>22.05</b>	33.46	44.83
AnyBURL <sup>‡</sup>	29.67	21.26	33.33	46.73	32.05	23.72	35.45	50.46	22.77	15.10	25.44	38.91
TLogic <sup>†</sup>	43.04	<u>33.56</u>	48.27	<u>61.23</u>	46.97	36.21	<u>53.13</u>	<u>67.43</u>	29.82	20.54	<u>33.95</u>	<u>48.53</u>
TECHS	<b>43.88</b>	<b>34.59</b>	<b>49.36</b>	<b>61.95</b>	<b>48.38</b>	<b>38.34</b>	<b>54.69</b>	<b>68.92</b>	<b>30.85</b>	<b>21.81</b>	<b>35.39</b>	<b>49.82</b>

Table 2: The experiment results (%) in ICEWS14, ICEWS0515 and ICEWS18. The optimal and suboptimal values of each metric are marked in bold and underlined respectively. Results of “<sup>‡</sup>” are from Liu et al. (2022), “<sup>†</sup>” means the results are from its original paper and others are all from Han et al. (2021).

Model	WIKI		YAGO	
	MRR	H@10	MRR	H@10
TTransE	29.27	42.39	31.19	51.21
TA-DistMult	44.53	51.71	54.92	66.71
DE-SimplE	45.43	49.55	54.91	60.17
TNTComplEx	45.03	52.03	57.98	66.69
CyGNet	33.89	41.86	52.07	63.77
RE-Net	49.66	53.48	58.02	66.29
xERTE	71.14	79.01	84.19	89.78
TITer	<u>75.50</u>	<u>79.02</u>	<u>87.47</u>	<u>90.27</u>
TECHS	<b>75.98</b>	<b>82.39</b>	<b>89.24</b>	<b>92.39</b>

Table 3: The experiment results (%) in WIKI and YAGO. The baseline results are from Sun et al. (2021).

Ablation	ICEWS14		ICEWS0515		ICEWS18	
	MRR	H@10	MRR	H@10	MRR	H@10
TECHS	43.88	61.95	48.38	68.92	30.85	49.82
w/o time	43.44	60.74	47.61	67.16	30.11	48.96
$\Delta$	0.44	1.21	0.77	1.76	0.74	0.86
w/o emd	42.45	60.21	46.57	66.68	29.87	48.34
$\Delta$	1.43	1.74	1.81	2.24	0.98	1.48
w/o PR	42.57	58.41	46.1	65.36	28.84	46.93
$\Delta$	1.31	3.54	2.28	3.56	2.01	2.89
w/o FO	42.84	60.06	46.27	65.49	29.78	47.59
$\Delta$	1.04	1.89	2.11	3.43	1.07	2.23

Table 4: The ablation results (%). *PR* and *FO* denote propositional and first-order respectively.

ble 2 and Table 3. As shown, our TECHS has achieved advanced performance. Compared with static embedding and temporal embedding models, e.g., the strongest RE-Net, our metrics have been greatly improved by 5.6%, 5.91%, 8.02% and 7.43% in ICEWS14. The performance of TECHS is also competitive with the explainable reasoning methods. It outperforms xERTE, TITer and AnyBURL by 3.09%, 2.15% and 14.21% MRR in ICEWS14, respectively. It demonstrates TECHS makes up for the shortcomings of simply using

propositional reasoning or static first-order logical rules on TKGs. Finally, compared with the state-of-the-art TLogic, TECHS also shows certain improvements, i.e., achieving better performance on all twelve metrics of ICEWS14, ICEWS0515 and ICEWS18 datasets. TECHS has an average improvement of 0.92%, 1.65% and 1.26% on these three datasets. Besides, TECHS yields 0.48%, 3.37%, 1.77% and 2.12% improvements in MRR and Hits@10 metrics in WIKI and YAGO datasets, compared with the state-of-the-art TITer. In summary, the results show the superiority of our model that conducts temporal graph embedding as well as integrates propositional and first-order reasoning.

### 5.3 Ablation Studies

To verify the effectiveness of each module in TECHS, ablation studies are carried out in Table 4. For “w/o time”, we remove the time embedding in the GCN. “w/o emd” means we remove the whole GCN encoder module and perform random initialization for embeddings. For the logical decoder, “w/o PR” or “w/o FO” means that we remove propositional or first-order attention in Eq. 8 when computing nodes’ attention for the ablation of the corresponding reasoning pattern. We analyze the results from the following two aspects: First, both topology structures and time dynamics in GCN embeddings contribute to extrapolation reasoning. When only removing time information, the metrics decrease slightly compared with the whole GCN ablation, e.g., 0.44% vs. 1.43% MRR drops in ICEWS14. Second, for logical reasoning, both propositional and first-order logic reasoning

is important. Propositional reasoning has a bigger impact in ICEWS14 than first-order reasoning (3.54% vs. 1.89% Hits@10 drops), while they have roughly the same effect in ICEWS0515 and ICEWS18 (3.56% vs. 3.43%, 2.89% vs. 2.23% Hits@10 drops). This may be due to the different topology structures of different datasets, resulting in different logical reasoning patterns. In summary, ablation studies show that structural dependencies and temporal dynamics as well as propositional and first-order reasoning all bring positive gains.

## 5.4 Hyperparameter Analysis

We run our model with different hyperparameters to explore weight impacts in Figure 4. Figure 4a shows the changes in the performance of models with different sampling hyperparameters  $M$  and  $N$ , where small values would lead to great performance decline. This is because fewer nodes and edges lead to insufficient and unstable training, respectively. When increasing  $M$  and  $N$ , the GPU memory of the model will increase rapidly in Figure 4b, especially for  $M$ . We also record the average training time of one epoch with different  $M$  and  $N$  in Figure 4c. Its overall trend is consistent with Figures 4a and 4b. In general, TECHS is time efficient as the running time is between 0.2 and 1 hour.

Figure 4d shows the impact of different weights when using first-order reasoning, where smaller weights show worse results, generally. Thus, the FOTH rule is functional for extrapolation reasoning on TKGs. Different contextualized, e.g., vanilla RNN, GRU, LSTM (Hochreiter and Schmidhuber, 1997) for FOTH rule learning and reasoning are compared in Figure 4e, where GRU outperforms the other two competitors. RNN performs worst, showing that simple models are not competent enough for discrete structures of FOTH rules.

To explore the effects of decoder iterations on model performance, we carry out experiments with iteration  $L=1, 2, 3, 4$  in ICEWS14, ICEWS0515 and ICEWS18. As Figure 4f shows, the performance generally improves with the iteration increasing. The metrics of  $L=3$  and  $L=4$  are similar, which shows that the answer is usually in the adjacent hops of the target entity. Larger hops bring more candidates, which may affect model performance, e.g., Hits@10 values drop when  $L=4$  in ICEWS14 and ICEWS18. Therefore,  $L=3$  is selected as the default setting in our experiments.

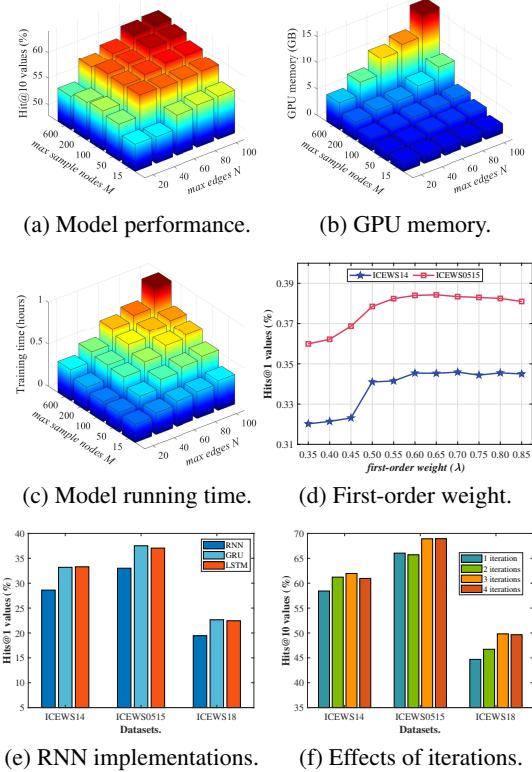


Figure 4: The effects of hyperparameters. (a), (b) and (c) are the effects of  $M$  and  $N$  on performance, GPU memory and training time. (d) denotes the impacts of the first-order weight. (e) and (f) show results with different RNN implementations and decoder iterations.

## 5.5 Case Study for Explainable Reasoning

Figure 5 visualizes two reasoning graphs on ICEWS14 and ICEWS0515, showing the extrapolation reasoning process of TECHS. The propositional attention weights of nodes are listed nearby them, which represent the propositional reasoning score of each node at the current step. For example, the uppermost propositional reasoning path from *Massoud Barzani* to *Iran: 2014-08-26* in case B learned a large attention score for the correct answer *Iran*. Generally, nodes with more prior neighbors or larger prior attention weights significantly impact subsequent steps and the prediction of final entity scores. From both reasoning cases, we induce several FOTH rules using the FARI algorithm. Some typical ones with their confidence scores are shown in Table 5. For example, the rule [7] with lower confidence is learned for the prediction of the false candidate *Iraq* in case B. These attentions and FOTH rules demonstrate the explainability of our model. Besides, we observe that propositional and first-order reasoning have an

	No.	$\epsilon$	premise $\Rightarrow$ conclusion
case A	[1]	0.22	$makeAppeal(X, Y_1):t_1 \wedge consult^{-1}(Y_1, Y_2):t_2 \wedge makeStatement(Y_2, Z):t_3 \Rightarrow appealCooperation(X, Z):t$
	[2]	0.13	$hostVisit^{-1}(X, Y_1):t_1 \wedge signAgreement(Y_1, Y_2):t_2 \wedge praise(Y_2, Z):t_3 \Rightarrow appealCooperation(X, Z):t$
	[3]	0.06	$expressIntentTo(X, Y_1):t_1 \wedge expressIntentTo(Y_1, Y_2):t_2 \wedge makeStatement(Y_2, Z):t_3 \Rightarrow appealCooperation(X, Z):t$
case B	[4]	0.17	$demand(X, Y_1):t_1 \wedge makeStatement(Y_1, Y_2):t_2 \wedge engageCooperation^{-1}(Y_2, Z):t_3 \Rightarrow makeStatement(X, Z):t$
	[5]	0.16	$consult(X, Y_1):t_1 \wedge expressIntentTo^{-1}(Y_1, Y_2):t_2 \wedge consult^{-1}(Y_2, Z):t_3 \Rightarrow makeStatement(X, Z):t$
	[6]	0.10	$demand(X, Y_1):t_1 \wedge consult(Y_1, Y_2):t_2 \wedge makeStatement(Y_2, Z):t_3 \Rightarrow makeStatement(X, Z):t$
	[7]	0.04	$praise(X, Y):t_1 \wedge makeStatement(Y, Z):t_2 \Rightarrow makeStatement(X, Z):t$

Table 5: Some FOTH rules learned during the reasoning process correspond to two cases in Figure 5. Existing signs ( $\exists$ ) are omitted for better exhibition and relations marked with red represent the target relation to be predicted.

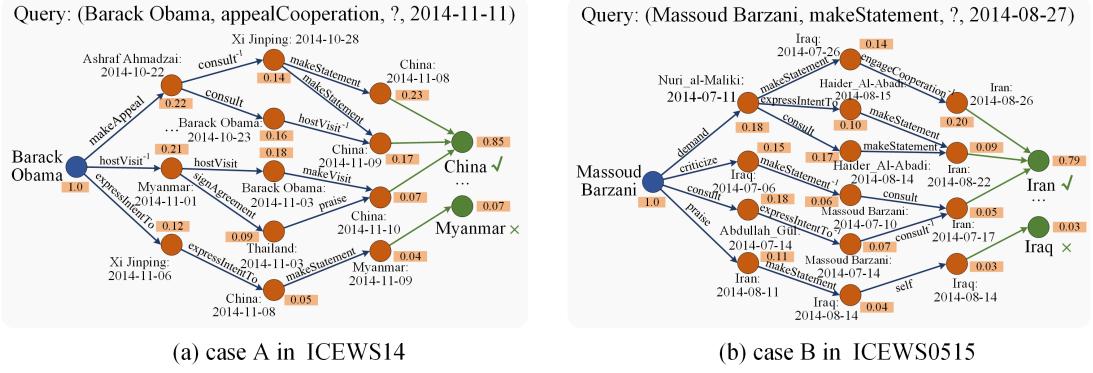


Figure 5: Cases of the reasoning processes, where values in orange rectangles represent propositional attentions.

incompletely consistent effect. Thus, they can be integrated to jointly guide the reasoning process, leading to more accurate reasoning results.

## 6 Conclusion

To effectively integrate complex information on TKGs and implement differentiable logical reasoning, this work proposes TECHS which mainly contains a temporal graph encoder and a logical decoder. The former utilizes the temporal encoding and heterogeneous attention to embed structural dependencies and temporal dynamics. The latter realizes differentiable rule learning and reasoning by continuously conducting forward message-passing in the proposed reasoning graph. Finally, FOTH rules can be easily induced by a novel FARI algorithm. In the future, we will explore mining more types of rules on TKGs, such as numerical rules (Wang et al., 2020), and expand to the scenario of inductive reasoning (Pan et al., 2022).

## 7 Limitations

Due to the massive combination of relations and times on TKGs, balancing the model performance and efficiency is challenging. Our model TECHS performs well as Section 5.2 and 5.4 discussed. However, there is also a limitation. TECHS is a

two-step approach that can be further improved if we can fuse logical reasoning in the graph encoder like ConGLR (Lin et al., 2022). The model will be more efficient for computational space and time.

## Acknowledgments

This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293553, 62250066, 62176207, 62192781, and 62250009), Innovative Research Group of the National Natural Science Foundation of China (61721002), “LENOVO-XJTU” Intelligent Industry Joint Laboratory Project, Foundation of Key National Defense Science and Technology Laboratory (6142101210201), Project of China Knowledge Centre for Engineering Science and Technology, Natural Science Basic Research Program of Shaanxi (2023-JC-YB-293), the Youth Innovation Team of Shaanxi Universities, XJTU Teaching Reform Research Project “Acquisition Learning Based on Knowledge Forest”.

## Ethical Statement

We honor the ethical code set out in the ACL Code of Ethics.

## References

- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. *Translating embeddings for modeling multi-relational data*. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2787–2795.
- Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2022. *Temporal knowledge graph completion: A survey*. *CoRR*, abs/2201.08236.
- Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. *Rlogic: Recursive logical rule learning from knowledge graphs*. In *The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 179–189. ACM.
- Junyoung Chung, Çağlar Gülcöhre, KyungHyun Cho, and Yoshua Bengio. 2014. *Empirical evaluation of gated recurrent neural networks on sequence modeling*. *CoRR*, abs/1412.3555.
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. *Fast rule mining in ontological knowledge bases with AMIE+*. *The VLDB Journal*, 24(6):707–730.
- Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2013. *AMIE: association rule mining under incomplete evidence in ontological knowledge bases*. In *22nd International World Wide Web Conference (WWW)*, pages 413–422. ACM.
- Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. *Learning sequence encoders for temporal knowledge graph completion*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4816–4821.
- Rishab Goel, Seyed Mehran Kazemi, Marcus A. Brubaker, and Pascal Poupart. 2020. *Diachronic embedding for temporal knowledge graph completion*. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 3988–3995.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. *Explainable subgraph reasoning for forecasting on temporal knowledge graphs*. In *9th International Conference on Learning Representations (ICLR)*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. *Long short-term memory*. *Neural computation*, 9(8):1735–1780.
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. *A survey on knowledge graphs: Representation, acquisition, and applications*. *TNNLS*, 33(2):494–514.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. *Recurrent event network: Autoregressive structure inference over temporal knowledge graphs*. In *EMNLP*, pages 6669–6683.
- Diederik P. Kingma and Jimmy Ba. 2015. *Adam: A method for stochastic optimization*. In *3rd International Conference on Learning Representations (ICLR)*.
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. *Tensor decompositions for temporal knowledge base completion*. In *8th International Conference on Learning Representations (ICLR)*.
- Julien Leblay and Melisachew Wudage Chekol. 2018. *Deriving validity time in knowledge graph*. In *Companion of the Web Conference (WWW)*, pages 1771–1776. ACM.
- Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. *Reasoning over different types of knowledge graphs: Static, temporal and multi-modal*. *CoRR*, abs/2212.05767.
- Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. *Incorporating context graph with logical reasoning for inductive relation prediction*. In *The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 893–903.
- Qika Lin, Rui Mao, Jun Liu, Fangzhi Xu, and Erik Cambria. 2023. *Fusing topology contexts and logical rules in language models for knowledge graph completion*. *Information Fusion*, 90:253–264.
- Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. *Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs*. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, pages 4120–4127. AAAI Press.
- Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. *YAGO3: A knowledge base from multilingual wikipedias*. In *Seventh Biennial Conference on Innovative Data Systems Research*.
- Rui Mao, Xiao Li, Mengshi Ge, and Erik Cambria. 2022. *MetaPro: A computational metaphor processing model for text pre-processing*. *Information Fusion*, 86–87:30–43.
- Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. *Anytime bottom-up rule learning for knowledge graph completion*. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3137–3143.
- Sean P O’brien. 2010. *Crisis early warning and decision support: Contemporary approaches and thoughts on future research*. *International Studies Review*, 12(1):87–104.
- Yudai Pan, Jun Liu, Lingling Zhang, Tianzhe Zhao, Qika Lin, Xin Hu, and Qianying Wang. 2022. *Inductive relation prediction with logical reasoning using contrastive representations*. In *Proceedings of the*

- 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4261–4274.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035.
- Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *9th International Conference on Learning Representations (ICLR)*.
- Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7710–7720.
- Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: end-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15321–15331.
- Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8306–8319.
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, volume 48, pages 2071–2080.
- L. Vivek Harsha Vardhan, Guo Jia, and Stanley Kok. 2020. Probabilistic logic graph attention networks for reasoning. In *Companion of The 2020 Web Conference*, pages 669–673. ACM / IW3C2.
- Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J. Smola, and Zheng Zhang. 2019. Deep graph library: Towards efficient and scalable deep learning on graphs. *CoRR*, abs/1909.01315.
- Po-Wei Wang, Daria Stepanova, Csaba Domokos, and J. Zico Kolter. 2020. Differentiable learning of numerical rules in knowledge graphs. In *8th International Conference on Learning Representations (ICLR)*.
- Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853.
- Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. In *8th International Conference on Learning Representations (ICLR)*.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations (ICLR)*.
- Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2319–2328.
- Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5773–5784.
- Wen Zhang, Jiaoyan Chen, Juan Li, Zezhong Xu, Jeff Z. Pan, and Huajun Chen. 2022b. Knowledge graph reasoning with logics and embeddings: Survey and perspective. *CoRR*, abs/2202.07412.
- Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Iteratively learning embeddings and rules for knowledge graph reasoning. In *The World Wide Web Conference (WWW)*, pages 2366–2377.
- Yao Zhang, Hongru Liang, Adam Jatowt, Wenqiang Lei, Xin Wei, Ning Jiang, and Zhenglu Yang. 2021. GMH: A general multi-hop reasoning model for KG completion. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3437–3446.
- Cunchao Zhu, Muahao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 4732–4740.
- Yifan Zhu, Qika Lin, Hao Lu, Kaize Shi, Donglei Liu, James Chambua, Shanshan Wan, and Zhendong Niu. 2023. Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 35(4):4178–4189.

---

## ACL 2023 Responsible NLP Checklist

### A For every submission:

A1. Did you describe the limitations of your work?

7

A2. Did you discuss any potential risks of your work?

4.3

A3. Do the abstract and introduction summarize the paper's main claims?

1

A4. Have you used AI writing assistants when working on this paper?

*Left blank.*

### B Did you use or create scientific artifacts?

*Left blank.*

B1. Did you cite the creators of artifacts you used?

*No response.*

B2. Did you discuss the license or terms for use and / or distribution of any artifacts?

*No response.*

B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?

*No response.*

B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?

*No response.*

B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?

*No response.*

B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.

*No response.*

### C Did you run computational experiments?

5

C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

5.4

---

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

5.1

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*No. We follow the same experimental setting and result presentation of previous studies for equal comparison. We run five times with different random seeds and report their mean results for each dataset.*

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*No. We did not use such packages.*

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*No response.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*No response.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*No response.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*No response.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*No response.*