Self-explainable Reasoning over Temporal Knowledge Graph with Adaptive Logical Rules

Qing Li, Guanzhong Wu*, **Jingjing Song, Chenglie Du**

School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China
{qingli, ducl}@nwpu.edu.cn, {wgz635035, songjingjing}@mail.nwpu.edu.cn

Abstract

The objective of temporal knowledge graph reasoning task is to predict future events based on historical factual knowledge. Existing methods for temporal knowledge graph reasoning typically fall into two categories that heavily depend on historical frequency information of entities or relations and lack flexibility in reusing historical information. To tackle these challenges , we propose **SEALR**, a **S**elf-**E**xplainable method for temporal knowledge graph reasoning with **A**daptive **L**ogical **R**ules to forecast future events with comprehensive and flexible historical information. SEALR learns representations of relations to mine semantic and sequential information within relation paths. Meanwhile, a time-aware encoder is proposed to capture the temporal information of relation paths. Through the fusion of historical information, SEALR can thoroughly investigate the logical correlations between relation paths and adaptively construct high-quality logical rules. Furthermore, a rule retrieval mechanism is proposed to flexibly apply the stored temporal logical rules. The experimental results on the three benchmark datasets demonstrate that our SEALR achieves superior performance while providing accurate and reasonable explanations.

Introduction

Temporal Knowledge Graphs (TKG) describe factual knowledge in the format of quadruplets, *i.e.*, *(subject,relation,object,timestamp)*, where timestamp signifies the specific occurrence time of an event. By being divid into a series of historical subgraphs along the timestamps, temporal knowledge depict historical dependencies and evolving trends of factual knowledge. As illustrated in Figure.1(a), the relationship between *Vietnam* and *China* evolves continuously and undergoes significant changes over several months.

TKG reasoning (TKGR) aims to capture patterns of event evolution and forecast future events at specific times. Given a TKG with T timestamps, TKGR task strives to predict the future links between entities after T. Such forecasting is particularly valuable in diverse domains like traffic flow prediction (Gu et al. 2023), financial analysis (Yu et al. 2023) and social network analysis(Wang et al. 2022).

*Corresponding author.

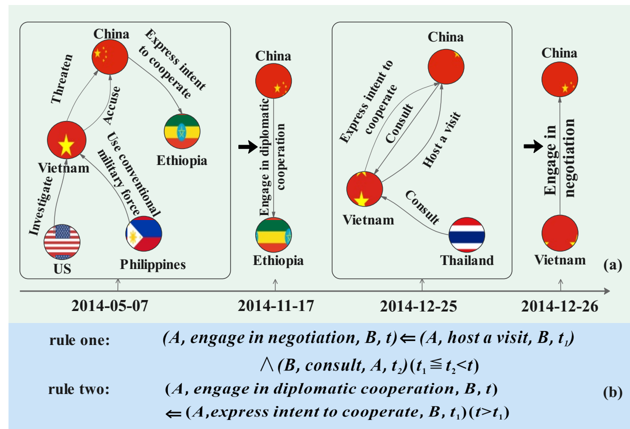

Figure 1: (a) Illustration of a TKG and the extrapolation reasoning over it. (b) An instance of two temporal logic rules extracted from figure (a).

In recent years, significant efforts have been dedicated to TKGR. Most of them are representation-based methods, they learn temporal embeddings of entities and relations from series of historical subgraphs with graph neural networks (GNN) or diverse vector scoring functions. Some other methods (Li, Sun, and Zhao 2022; Zhang et al. 2023b,a; Liang et al. 2023) learn the representations by aggregating intra-structural information over subgraphs and evolving inter-temporal information between subgraphs. Since these black-box models lack explainability for predictions, some rencent methods (Liu et al. 2022; Bai et al. 2023c,a) attempt to apply logical rules for TKGR task. They extract temporal logical rules from multi-hop paths within historical subgraphs and evaluate the confidence of the rules for application. While these methods achieve promising results, they face challenges in predicting unseen entities, which is a significant and widespread issue in TKG reasoning (Xu et al. 2023). Unseen entities refer to those who have minimal interaction traces in historical subgraphs. Existing methods face the following two primary limitations in predicting unseen entities.

Firstly, existing models heavily rely on historical frequency information. Most of representation-based models utilize relational graph convolutional network (RGCN) and its variants to aggregate neighbor information, potentially overlooking low-frequency interactions. For instance, *Vietnam* is more likely to be predicted for query *(China,Engage in diplomatic cooperation,?,2014-11-17)* in Figure.1(a) because of its higher interaction frequency with *China* compared to *Ethiopia*. Contrastingly, models employing temporal logical rules mitigate this issue by rule matching to alleviate it during the prediction phase. The correct answer *Ethiopia* can be retrieved with temporal logical rule one in Figure.1(b), even if there is minimal interaction. However, temporal logical rules are typically evaluated and determined based on the frequency of relation paths in previous work, which may lead to the omission of crucial rules.

Secondly, existing models lack flexibility in modeling historical facts over long time spans. For representation-based models, they primarily utilize GRU to evolve embeddings of entities and relations along the series of timestamps. While GRU may suffer from information forgetting when modeling long sequences, leading to the loss of historical information over long time spans. This issue also exists in rule-based models, where learning and applying temporal logical rules over long time spans also introduce significant additional computational overhead and redundant information. Furthermore, existing methods usually focus on fixed-length historical information, whereas for each query, the valuable historical information may have a flexible and variable time span.

To alleviate the above limitations, we propose a **S**elf-**E**xplainable method for temporal knowledge graph reasoning with **A**daptive **L**ogical **R**ules (SEALR), which comprehensively and flexibly capture potential temporal logical rules in TKGs. SEALR introduces embeddings of relations and extracts temporal logical rules based on semantic associations between vectors. Specifically, we encode historical relation paths using embeddings of relationships and timestamps and evaluate their logical correlations with the query relation based on semantic similarity. This allows SEALR to iteratively search for possible temporal logical rules from historical facts. In addition, confidence assessment based on semantic similarity avoids excessive reliance of logical rules on frequency information. To flexibly learn and apply temporal logical rules from historical subgraphs of arbitrary time spans, we define a dynamic and adjustable time-aware encoder. After learning new temporal logical rules, we save them to the rule memory. By flexibly applying temporal logical rules, SEALR comprehensively captures historical information, including unseen entities and their interactions, leading to improved performance in forecasting future events. The main contributions of our work are summarized as follows:

- We propose a self-explainable reasoning approach for TKGR task with adaptive logical rules, termed SEALR, which adaptively extract potential logical rules to effectively mitigate the issue of unseen entities in TKGs.
- A novel time-aware encoder and rule retrieval mechanism are developed, enabling our method to utilize historical information with low frequency and arbitrary time spans to learn and apply the temporal logic rules.
- Extensive experiments on three datasets demonstrates the comprehensive performance of our SEALR, including superiority, effectiveness, sensitivity, and explainability.

Related work

Representation-based methods for TKGR

To tackle the TKGR task, some works have endeavored to adapt graph representation learning methods with temporal information. They integrate timestamp embeddings into the scoring function of graph embedding models to introduce temporal dependencies between entity and relation embeddings, such as TTransE (Leblay and Chekol 2018), TNTComplEx (Lacroix, Obozinski, and Usunier 2020) and TA-DisMult (García-Durán, Dumancic, and Niepert 2018). However, their lack of scalability limits their performance in forecasting future links. Recently, some methods (Jin et al. 2020a; Li et al. 2021; He et al. 2021; Li, Sun, and Zhao 2022; Park et al.; Zhang et al. 2023c) focus on learning the representations by aggregating intra-structural information over subgraphs with various GNN and evolving inter-temporal information between subgraphs with GRU. By uncovering the evolution patterns of entity and relation representations within historical sequences with GRU, they achieve event prediction for future timestamps. Nevertheless, these methods fail to provide reasonable explanations for their reasoning results.

Path-based methods for TKGR

As the multi-subgraph structure of TKGs naturally lends itself to multi-hop reasoning, some methods (Sun et al. 2021; Bai, Chai, and Zhu 2023; Zheng et al. 2023) attempts to utilize reinforcement learning to search for target entities and provide reasoning processes through paths. Meanwhile, some methods attempt to extract relation paths from historical multi-hop paths and define them as logical rules. For example, TLogic (Liu et al. 2022) collects temporal logic rules on TKGs with a random walk and calculates confidence based on the frequency of facts. TILP (Xiong et al. 2023) designs a constrained random walk mechanism and the introduction of temporal operators to improve the reasoning performance. These methods effectively utilize statistically-based confidence assessment to identify temporal logical rules. However, they are limited to extracting logical rules from provided historical data and cannot iteratively update them in continuously evolving temporal knowledge graphs. Besides, they heavily rely on frequency information and lack flexibility when training and prediction.

Preliminaries

Temporal Knowledge Graph

Let $\mathcal{E}$ denote the set of entities, $\mathcal{R}$ the set of relations, and $\mathcal{T}$ the set of timestamps. A temporal knowledge graph can be formalized as $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_{t-1}, \mathcal{G}_t\}$, where $\mathcal{G}_t$ is the historical subgraph corresponding to the timestamp $t \in \mathcal{T}$. $\mathcal{G}_t$ consists of facts at the current timestamp, where each fact is formalized as quadruplets (s, r, o, t). The subject entity $s \in$

$\mathcal{E}$ and the object entity $o \in \mathcal{E}$ are linked by relation $r \in \mathcal{R}$. To extend the connectivity of the graph, we add an inverse edge (s, r^{-1}, o, t) to each edge (s, r, o, t) in the graph $\mathcal{G}$. The units of timestamps vary depending on the requirements and could be in years, months, or days.

TKG reasoning is a link prediction task for future timestamps. Given a query quadruplet $(s, r, ?, t_q)$, we attempt to identify candidate entities based on historical subgraph sequences $\mathcal{G}_{t<t_q}$ and rank them according to their likelihood.

Temporal Logical Rule

With the presence of multiple paths on the knowledge graph (Niu and Li 2023), first-order logical rules are naturally suitable for link prediction tasks. When applied to the temporal knowledge graph, a logical rule can be formalized as:

$$\begin{aligned}(E_s, R_h, E_o, T_h) \leftarrow (E_{n-1}, R_n, E_o, T_n) \wedge \\ (E_{n-2}, R_{n-1}, E_{n-1}, T_{n-1}) \wedge \cdots \wedge (E_s, R_1, E_1, T_1),\end{aligned} \tag{1}$$

where the left side of the equation is the rule head, while the right side of the equation is the rule body. It is worth noting that

$$T_h > T_n \geqq T_{n-1} \geqq \cdots \geqq T_1, \tag{2}$$

is the temporal constraint, which ensures the validity of the rules. The length of the rules is variable, so there can be multiple rules for a single rule head.

To mine the latent relation patterns, we extract relation paths $\{R_1, R_2, ..., R_n\}$ as the rule body with the query relation R_h as the rule head. Then the temporal logic rules can be replaced with:

$$(R_h, T_h) \leftarrow (R_n, T_n) \wedge (R_{n-1}, T_{n-1}) \wedge \cdots \wedge (R_1, T_1). \tag{3}$$

We regard Eq.(3) as a set of temporal relation patterns. In contrast to TLogic, we calculate the similarity between the rule body and the rule head as confidence of a rule, rather than frequency.

METHODOLOGY

As illustrated in Figure.2, there are three main components in SEALR: (1) *Historical Logical Path Extraction*, which is to search for valid relation paths with temporal constraints from historical subgraphs; (2) *Temporal Rule Encoder*, which is to capture temporal and semantic information in the relation paths. (3) *Rule Construction*, which is to construct effective temporal logical rules based on confidence and memorize them. Finally, SEALR applies the rules to unknown timestamps to predict future events.

Historical Logical Path Extraction

To extract temporal logical rules from historical subgraphs, we initially search for valid logical reasoning paths that satisfy the temporal constraints in Eq.3. Specifically, for a query event (s, r_q, o, t_q), we commence by retrieving two k-hop subgraphs around nodes s and o from their historical interactions, respectively. We then intersect these two subgraphs to form a search graph $\mathcal{G}_{search}$, which encompasses all possible paths with a maximum length $L \leq 2k$ between subject s and object o. Additionally, we implement a variable time window to confine the visible history

Algorithm 1: Path Extraction

Input: the search graph $\mathcal{G}_s$, query quadruplet (s, r_q, o, t_q), max path length L, max path number M

Output: the list of valid extracted paths P

Init $P = Dict([])$; $q = Stack()$; $visited = List()$; $prev = Dict()$; $count = 0$
q.push(s); $visited[s] = True$
while q is not empty && $count < M$ **do**
  $u = q[-1]$
  if u not in $prev$.keys() **then**
    $prev[u] = List()$
  end if
  for v in $\mathcal{G}_s[u]$ **do**
    if $visited[v] == 0$ and v not in $prev[u]$ **then**
      q.push(v); $visited[v] = True$; $prev[u]$.push(v)
      if $v == o$ **then**
        $P[|q|]$.push(q); q.pop(); $visited[v] = 0$; $count += 1$
      end if
      if $|q| > L$ **then**
        $visited[v] == 0$; q.pop()
      end if
    end if
  end for
  if $|prev[u]| == 0$ **then**
    $d = q$.pop(); $prev[d]$.pop(); $visited[d] = 0$
  end if
end while

within a specific time range, considering that TKGs may encompass thousands of timestamps. To leverage static graph search algorithms, we mask the timestamps of quadruplets in $\mathcal{G}_{search}$ while preserving the basic graph structure. After that, we simultaneously apply depth-first and breadth-first search strategies within graph $\mathcal{G}_{search}$ while constraining the maximum path length L and the maximum number of paths M per query quadruplet. The implementation of the path extract algorithm is detailed in Algorithm 1.

For each path in $\mathcal{P}_{search}$, we apply it to the graph $\mathcal{G}_h$ to take the corresponding edges, which can be defined as follows:

$$p_e = \{(s, R_1, e_1, T_1), (e_1, R_2, e_2, T_2), \ldots, (e_{l-1}, R_l, o, T_l)\}, \tag{4}$$

where $l \leq L$ is the path length. $R_i \in \mathcal{R}$ denotes the set of relations between e_{i-1} and e_i, while T_i is the corresponding timestamps. Then, the multi-hop path of relation sets with timestamps are extracted as

$$p_r = \{(R_1, T_1), ..., (R_l, T_l)\}, \tag{5}$$

Where p_r represents the relation paths with timestamps. We instantiate the sequence of relation sets into specific relation paths and remove those that do not satisfy the temporal constraints outlined in Eq.3. Finally, we obtain a set of temporal relation paths, denoted as

$$\mathcal{P}_r = \sum_{l=1}^{L} \{(r_1, t_1), (r_2, t_2), \ldots, (r_l, t_l)\}, \tag{6}$$

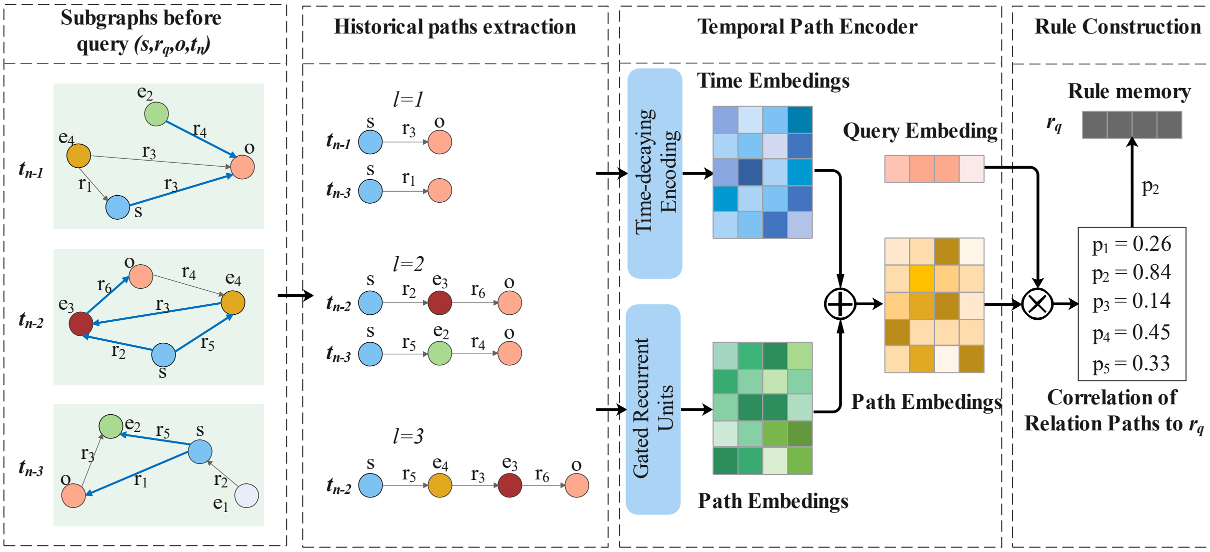

Figure 2: An overview of the proposed SEALR method.

where $\mathcal{P}_r$ denotes all the valid relation paths extracted from paths similar to p_r with varying lengths.

Temporal Path Encoder

After extracting the historical relation paths among historical subgraphs, we introduce a GRU and a time-aware encoder to encode the relation paths and their corresponding timestamps separately.

Relation path encoding The relation paths reflect the latent logical dependencies between relations. Therefore, utilizing GRU is beneficial for capturing sequential insights within these paths, enabling the extraction of latent features. The update mechanism of GRU can be succinctly summarized as follows:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t, \tag{7}$$

where h_t is the hidden representation at timestamp t, z_t is the update gate, h_{t-1} is the hidden state of the prior timestamp $t-1$, and h'_t is the information of current unit. To obtain hidden representations, we feed the relation paths into GRU as follows:

$$\mathbf{H}_r = GRU(\hat{\mathcal{P}}_r), \tag{8}$$

where $\hat{\mathcal{P}}_r = \mathcal{P}_r \backslash T$ is the set of relation paths without temporal information, $\mathbf{H}_r \in \mathbb{R}^{n\times d}$ signifies their corresponding embeddings.

Time-decaying encoder The timestamps play a crucial role in TKGs, indicating the occurrence time of facts and revealing patterns of recurrence and evolution of these facts. Events with similar timestamps often demonstrate closer associations, as do the relations within them. Building on this premise, we designed the following time-aware function:

$$\phi(\Delta t) = tanh(\Delta t^2/\gamma), \tag{9}$$

where Δt represents the time interval between two timestamps. γ is a parameter to control the gradient of the function. $\phi(t)$ represents the influence affected by the time interval t, which decreases as the interval is smaller. We can adjust the magnitude of γ to adapt to historical subgraphs of varying lengths.

We take T_1 as temporal attribute of the paths. Then, the time interval between each path and the query quadruplet is calculated by $|t_q - T_1|$. After that, we can obtain the representations of temporal information within each path by

$$\mathbf{H}_t = \mathbf{W}_t\phi(\tau) + \mathbf{b}, \tag{10}$$

where $\mathbf{W}_t \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^d$ are learnable parameters, $\tau = |r_q - T_1|$ is the time intervals. $\mathbf{H}_t \in \mathbb{R}^{n\times d}$ is the time embeddings of the relation paths.

Finally, we update the embeddings of relation paths by incorporating time embeddings to

$$\mathbf{H}_p = \mathbf{H}_r + \mathbf{H}_t. \tag{11}$$

The final representation $\mathbf{H}_p$ aggregates both the sequential and temporal information extracted from the relation paths, thereby enhancing its capability to capture the intrinsic correlation between the relation paths and the query relation.

Rule construction

For a given query quadruplet (s, r_q, o, t_q), we can extract a set of relation paths from the historical subgraphs. To construct credible temporal logical rules, we identify the most plausible path among them based on confidence.

Specifically, we estimate the confidence of these relation paths by calculating their semantic similarity with the query relation r_q. After the fusion of temporal and path embeddings, the embeddings of these paths reflect their temporal and semantic features. Therefore, potential logical associations can be discovered by computing the semantic similarity between embeddings using the cosine similarity:

$$sim(\mathbf{r_q}, \mathbf{h_i}) = \frac{\mathbf{r_q} \cdot \mathbf{h_i}}{||\mathbf{r_q}||_2 \cdot ||\mathbf{h_i}||_2}, \tag{12}$$

where $\mathbf{r_q}$ and $\mathbf{h_i}$ are embeddings of the query relation and relation paths, respectively.

Among the extracted relation paths $\mathcal{P}_r$, the path with exhibiting the highest similarity is deemed as the most reasonable. Therefore, we designate it as the candidate rule body for the query quadruplet and score the quadruplet based on their similarity:

$$score((s, r_q, o, t_q)) = \max_{p_i \in \mathcal{P}_r}\{sim(\mathbf{r_q}, \mathbf{h_i})\}. \quad (13)$$

To construct the final temporal logic rules, we further filter the paths based on the scores of quadruplets. We apply a threshold to eliminate paths with low confidence, thereby deriving the final temporal logical rules:

$$C = \sum_{i=1}^{|\mathcal{F}_{train}|} r_{q_i} \wedge p_i \quad if\ score(Q_i) > \varepsilon, \quad (14)$$

where ε is the threshold of confidence, $\mathcal{F}_{train}$ represents known facts, *i.e.*, the training set of a TKG. Q_i denotes the i-th query quadruplet in $\mathcal{F}_{train}$, and r_{q_i} is the query relation within it. $p_i = \{(r_1, t_1), ..., (r_l, t_l)\}$ is the relation path with the highest score for query Q_i. C represents the collection of all temporal logic rules captured from known facts.

For each pair of relation and relation path in C, we remove the timestamps and store them in the following format:

$$C_r = \sum_{j=1}^{|\mathcal{R}|}(r_j, \hat{p}_j), \quad (15)$$

where $|\mathcal{R}|$ is the number of all relations in a TKG, r_j and $\hat{p}_j = \{r_1, ..., r_l\}$ are the query relation and the corresponding relation path without timestamps respectively.

Training objective

During the training phase, our goal is to optimize the embeddings of relations to capture the most plausible logical rules. We introduce perturbations by replacing the subject or object entity of each quadruplet with unrelated nodes. These modified quadruplets are regarded as negative quadruplets, from which we extract incorrect relation paths. Motivated by (Bai et al. 2023b), we define the loss function as follows:

$$\mathcal{L}_1 = \begin{cases} 1 - score(Q) & y = 1 \\ max(0, score(Q) - \delta) & y = -1, \end{cases} \quad (16)$$

where $\delta \in (-1, 1)$ is the margin of confidence and $y \in \{-1, 1\}$ denotes the label. $score(Q)$ represents the highest confidence score of extracted relational paths corresponding to quadruple Q. Eq.(16) aims to enhance the confidence of the rules constructed from the positive quadruplets while decreasing the negative ones. The negative scores are limited to be less than δ, as the paths from negative quadruplets may exist in the TKG with other timestamps.

$\mathcal{L}_1$ effectively optimizes the representations of relations. However, its linear gradient allows the loss to decrease rapidly, potential resulting to local optima issues. To mitigate this problem, we introduce the following loss function:

$$\mathcal{L}_2 = \begin{cases} log(1 + \exp(1 - score(p))) & y = 1 \\ log(1 + \exp(1 + score(p))) & y = -1. \end{cases} \quad (17)$$

Similar to Eq.(16), Eq.(17) optimizes both positive and negative quadruplets simultaneously with more soft gradient.

Thus, the total loss function of SEALR is formalized as follows:

$$\mathcal{L} = \alpha\mathcal{L}_1 + (1 - \alpha)\mathcal{L}_2. \quad (18)$$

where α is a constant to adjust weights of the two losses.

Inferring with rules

To infer an unknown event $(s, r_q, ?, t_q)$ in the future, we first retrieve a set of historical subgraphs based on the time window m. The value of m is variable depending on different datasets. Then we retrieval the logical rules corresponding to r_q, which are memorized as Eq.(15). For each logical rules, we apply it to the historical subgraphs to extract valid multi-hop paths with temporal constraints. During the inference phase, each logical rule can be treated equally, regardless of the frequency of their occurrence during training. Subsequently, we obtain a collection of multi-hop paths containing candidate entities, timestamps, and relation paths. We encode timestamps and relational paths using Eq.(7)-Eq.(11) and score each candidate entity based on the semantic similarity of relation paths with the query relation. Notably, the same candidate entity may correspond to multiple relation paths; however, we only consider the highest-scoring path among them, which is also used to interpret the inference result. Finally, we rank all candidate entities by their scores to answer the query quadruplet $(s, r_q, ?, t_q)$, while providing the corresponding basis for interpretable reasoning.

Experimental Results

Datasets & Metrics We evaluate our SEALR on three real-world datasets in the experiments: ICEWS14 (García-Durán, Dumancic, and Niepert 2018), ICEWS18 (Jin et al. 2020b) and ICEWS05-15 (García-Durán, Dumancic, and Niepert 2018). The three datasets are extracted from the Integrated Crisis Early Warning System[1] (ICEWS), which contain the facts in 2014, 2018, and facts from 2005 to 2015, respectively. Following previous works (Jin et al. 2020a; Li et al. 2021; Zhang et al. 2023c), the datasets are divided into training, validation, and test sets with a ratio of 8:1:1 by timestamps. We evaluate our method with two widely-used metrics in knowledge graph completion, MRR and Hits@1,3,10. For a query, we predict its subject and object entities and report the average results.

Baselines To validate the performance of our SEALR, we compare it with **representation-based** and **Path-based** methods, including TTransE (Leblay and Chekol 2018), TA-DisMult (García-Durán, Dumancic, and Niepert 2018), TNTComplEx (Lacroix, Obozinski, and Usunier 2020), RE-NET (Jin et al. 2020a), RE-GCN (Li et al. 2021), HIP (He et al. 2021), TiRGN (Li, Sun, and Zhao 2022), EvoKG (Park et al.), HGLS (Zhang et al. 2023c), TITer (Sun et al. 2021), TLogic (Liu et al. 2022) and TECHS (Lin et al. 2023).

[1] https://dataverse.harvard.edu/dataverse/icews

Table 1: Comparison results(in percentage) of all methods on ICEWS datasets for TKGR task. The best performances are highlighted in boldface, and the underlined results represent the second-best.

Methods	ICEWS14				ICEWS18				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TTransE(2018)	12.86	3.14	15.72	33.65	8.44	1.85	8.95	22.38	16.53	5.51	20.77	39.26
TA-DisMult(2018)	26.22	16.83	29.72	45.23	16.42	8.60	18.13	32.51	27.51	17.57	31.46	47.32
TNT-ComplEx(2020)	32.12	23.35	36.03	49.13	21.23	13.28	24.02	36.91	27.54	19.52	30.80	42.86
RE-NET(2020)	35.77	25.99	40.10	54.87	26.17	16.43	29.89	44.37	36.86	26.24	41.85	57.60
RE-GCN(2021)	37.78	27.17	42.50	58.84	27.51	17.82	31.17	46.55	38.27	27.43	43.06	59.93
HIP(2021)	<u>50.57</u>	<u>45.73</u>	<u>54.28</u>	61.65	<u>48.37</u>	<u>43.51</u>	<u>51.32</u>	<u>58.49</u>	<u>52.76</u>	<u>46.35</u>	55.31	61.87
TiRGN(2022)	44.04	33.83	48.95	63.84	33.66	23.19	37.99	54.22	50.04	39.25	<u>56.13</u>	<u>70.71</u>
EvoKG(2022)	27.18	30.84	-	47.67	29.28	33.94	-	50.09	-	-	-	-
HGLS(2023)	47.00	35.06	-	<u>70.41</u>	29.32	19.21	-	49.83	46.21	35.32	-	67.12
TITer(2021)	40.90	31.77	45.84	57.67	28.44	20.06	32.07	44.33	46.62	36.46	52.29	65.23
Tlogic(2022)	41.80	31.93	47.23	60.53	28.41	18.74	32.71	47.97	45.99	34.49	52.89	67.39
TECHS(2023)	43.88	34.59	49.36	61.95	30.85	21.81	35.39	49.82	48.38	38.34	54.69	68.92
SEALR(2024)	**63.65**	**55.29**	**64.21**	**75.64**	**53.35**	38.89	**62.37**	**75.39**	**64.92**	**50.91**	**72.31**	**87.69**
+improve	13.08	9.56	9.93	0.23	4.98	-	11.05	16.90	12.16	4.56	15.20	15.94

Table 2: Ablation results of SEALR on the ICEWS14.

Model	MRR	Hits@1	Hits@3	Hits@10
SEALR-tv	60.53	47.56	70.35	81.09
SEALR-L1	45.39	31.40	54.88	70.73
SEALR-L2	33.17	23.17	37.19	56.40
SEALR-all	47.14	34.00	53.00	76.29
SEALR-top3	58.33	44.57	<u>69.43</u>	<u>85.29</u>
SEALR-top2	<u>61.50</u>	<u>47.57</u>	**72.86**	**86.14**
SEALR	**63.65**	**55.29**	64.21	70.64

Parameter settings Due to the various number of timestamps, we set the time window m of historical subgraphs as 5, 3, and 20 in ICEWS14, ICEWS18, and ICEWS05-15 when training. For testing, we set m to 300 for the three datasets. Correspondingly, γ is set to 360 for training and 3600 for testing. The maximum path length L and maximum number M for path extracting are set to 3 and 150, respectively. We set vector dimension $d = 200$ and batch size to the size of each timestamp for all the datasets. The value of weight parameter α is set to 0.5. We adopt the Adam for parameter learning and set the learning rate to 1e-5.

Comparison results

As illustrated in Table 1, our SEALR exhibits significant superiority over other methods on ICEWS datasets, with average performance improvements about 10.07%, 4.71%, 12.06% and 12.8% on four metrics, respectively. The state-of-the-art performance verifies the superiority of our method.

For representation-based TKGR models, traditional TKG embedding models fail to predict future events evolving from existing history, leading to poor performance. In contrast, recent models utilize neural networks to learn patterns of historical information changes, thereby enabling inference on future events. For example, HGLS (Zhang et al. 2023c) establishes virtual links across historical subgraphs, enabling RGCN to aggregate historical information over arbitrary spans. Nevertheless, HGLS incorporates a large amount of irrelevant information when aggregating historical information, which limits its performance. These models ignore the latent relation patterns within TKGs and heavily rely on the frequency information of historical interactions between entities. They lack the ability to further model unseen entities, which is one of the reasons why our SEALR is significantly ahead of them.

Path-based models provide explainability that the above models lack. TITer (Sun et al. 2021) employs reinforcement learning to search for target entities, whereas TLogic and TECHS utilize temporal logic rules to retrieve candidate entities. However, they solely determine the logical correlations between relations based on frequency rather than utilizing semantic correlations, leading to their neglect of low-frequency logical rules. Our SEALR outperforms them significantly because we focus on the frequency-independent correlation to mine the latent relation patterns. It is worth noting that SEALR particularly excels on ICEWS05-15, thanks to our time-aware functions and unique rule memory mechanism.

Ablation studies

To investigate the effectiveness of the different modules in SEALR, we design corresponding variants of SEALR and show their performances on ICEWS14.

Impact of time-aware function To demonstrate the superiority of the time-aware function, we replaced it with Time2Vec (Xu et al. 2020) as **SEALR-tv**:

$$\phi(t) := \sqrt{\frac{1}{d}}\left[\cos(\mathbf{w}_1 t + \mathbf{p}_1), \cdots, \cos(\mathbf{w}_d t + \mathbf{p}_d)\right], \qquad (19)$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{p} \in \mathbb{R}^d$ are learnable parameters. As shown in Table 2, the performance of **SEALR-tv** decreases compared to SEALR, which indicates that the proposed time-aware function can effectively encode temporal information. Besides, by adjusting γ, the time-aware function can adapt to datasets with varying numbers of timestamps.

Table 3: Cases of inference with SEALR on ICEWS. The candidate entities are highlighted in boldface.

Query quadruplet	No.	Historical path	Confidence
(a) (Ashraf Ghani Ahmadzai, Sign formal agreement^{-1}, China, 2014-11-02)	[1]	(Ashraf Ghani Ahmadzai, Express intent to meet or negotiate, Xi Jinping,2014-10-29) (Xi Jinping, Make statement, **China**, 2014-10-30)	0.6748
	[2]	(Ashraf Ghani Ahmadzai, Make statement^{-1}, Mohammad Javad Zarif, 2014-10-01) (Mohammad Javad Zarif, Make an appeal or request,**China**, 2014-10-20)	0.2399
	[3]	(Ashraf Ghani Ahmadzai, Make statement^{-1}, Abdullah Abdullah, 2014-09-22) (Abdullah Abdullah, Criticize or denounce, **Head of Government (Afghanistan)**, 2014-10-03)	0.1544
(b) (Boko Haram, Abduct, hijack, or take hostage, Citizen (Nigeria), 2014-11-04)	[4]	(Boko Haram, Make statement,Militant (Boko Haram),2014-05-14) (Militant (Boko Haram), Abduct, hijack, or take hostage, **Citizen (Nigeria)**, 2014-05-16)	0.3059
	[5]	(Boko Haram,Criticize or denounce, **Citizen (Nigeria)**, 2014-05-09)	0.1526
	[6]	(Boko Haram, Make statement^{-1}, Kashim Shettima, 2014-08-20) (Kashim Shettima, Praise or endorse, **Government (Nigeria)**, 2014-08-23)	0.1065

Impact of loss functions We designed two variants of SEALR to validate the necessity of jointly training with two loss functions. **SEALR-L1** denotes training with only $\mathcal{L}_1$, and **SEALR-L2** refers to training with only $\mathcal{L}_2$. From the experimental results in Table 2, it is obvious that the model's performance significantly deteriorates when either loss function is used in isolation. When trained in conjunction, $\mathcal{L}_1$ accelerates the model's convergence, while $\mathcal{L}_2$ alleviates the overfitting issue introduced by $\mathcal{L}_1$. Therefore, both of the loss functions are indispensable, verifying the effectiveness of our method.

Impact of rule memory mechanism To verify the effectiveness of the rule memory mechanism we proposed, we compare SEALR with three variants: **SEALR-all**, which calculates scores for all candidates when applying rules to infer future events; **SEALR-top3**, which retains only the top three candidates; **SEALR-top2**, which retains only the top two candidates.

It can be observed that SEALR, which infers with only top one candidates, outperforms these variants, reflecting the impact of frequency information on SEALR's performance. In the settings of **SEALR-all**, entities with more historical interactions receive higher scores. **SEALR-all** outperforms previous path-based models, indicating that our model extracts higher-quality rules during the training phase. As the influence of frequency information decreases, the model's performance gradually improves from **SEALR-top3** to **SEALR-top2**, providing strong evidence that latent logical correlation between relations are the key to mine the logical rules. It's worth noting that **SEALR-top2** exceeds SEALR on Hits@3 and Hits@10, indicating that frequency information does indeed play an important role in prediction, which is the core of previous work.

Case study

Table 3 displays some examples of SEALR's inference in the reasoning process on the ICEWS14. Given a query quadruple with a future timestamp, SEALR can provide explainable reasoning paths with different lengths and temporal spans. For a query relation, SEALR saves numerous corresponding relation patterns through the rule memory mechanism, which can generate corresponding paths in historical subgraphs. The temporal spans of these paths can be quite extensive, as seen in, such as path [4] and [5]. Nevertheless, paths [4] and [5] have higher confidence than path [6], thereby achieving predictions for the correct object entity. This is because we employ a relevance-based approach to calculate the confidence of relation paths, mitigating the influence of time and frequency information. We only utilize the highest confidence paths, i.e., paths [1] and path [4], to predict future events. The paths not only enhance the prediction accuracy of SEALR, but more importantly, they provide time-consistent reasoning basis for explainability. In fact, both paths [1] and [4] provide reasonable explanations for inferring the object entities.

Conclusion

In this paper, we propose a novel self-explanatory method called SEALR for reasoning with temporal knowledge graphs and latent relation patterns. Firstly, our method extracts temporal paths from historical subgraphs to generate temporal logic rules. We introduce a time-aware function that filters paths based on their higher temporal relevance. To construct the temporal logic rules and preserve them as relation patterns, we evaluate the relevance of paths using semantic similarity. Furthermore, we develop a novel rule memory mechanism that enables us to store and use the rules for explainable reasoning. We conduct extensive experiments on three benchmarks to demonstrate the effectiveness and explainability of SEALR. Additionally, by incorporating relation patterns, SEALR is able to predict events over long temporal spans and low-frequency occurrences, effectively addressing the issue of unseen entities in temporal knowledge graph reasoning.

Acknowledgments

This work is supported in part by the National Science Fund for Distinguished Young Scholars of China (62325308).

References

Bai, L.; Chai, D.; and Zhu, L. 2023. RLAT: Multi-hop Temporal Knowledge Graph Reasoning Based on Reinforcement Learning and Attention Mechanism. *Knowledge-Based Systems*, 269: 110514.

Bai, L.; Chen, M.; Zhu, L.; and Meng, X. 2023a. Multi-Hop Temporal Knowledge Graph Reasoning with Temporal Path Rules Guidance. *Expert Systems with Applications*, 223: 119804.

Bai, L.; Chen, M.; Zhu, L.; and Meng, X. 2023b. Multi-hop temporal knowledge graph reasoning with temporal path rules guidance. *Expert Syst. Appl.*, 223: 119804.

Bai, L.; Yu, W.; Chai, D.; Zhao, W.; and Chen, M. 2023c. Temporal knowledge graphs reasoning with iterative guidance by temporal logical rules. *Inf. Sci.*, 621: 22–35.

García-Durán, A.; Dumancic, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 4816–4821. Association for Computational Linguistics.

Gu, B.; Zhan, J.; Gong, S.; Liu, W.; Su, Z.; and Guizani, M. 2023. A Spatial-Temporal Transformer Network for City-Level Cellular Traffic Analysis and Prediction. *IEEE Trans. Wirel. Commun.*, 22(12): 9412–9423.

He, Y.; Zhang, P.; Liu, L.; Liang, Q.; Zhang, W.; and Zhang, C. 2021. HIP Network: Historical Information Passing Network for Extrapolation Reasoning on Temporal Knowledge Graph. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 1915–1921. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-9-6.

Jin, W.; Qu, M.; Jin, X.; and Ren, X. 2020a. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6669–6683. Association for Computational Linguistics.

Jin, W.; Qu, M.; Jin, X.; and Ren, X. 2020b. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 6669–6683. Association for Computational Linguistics.

Lacroix, T.; Obozinski, G.; and Usunier, N. 2020. Tensor Decompositions for Temporal Knowledge Base Completion. In *International Conference on Learning Representations*.

Leblay, J.; and Chekol, M. W. 2018. Deriving Validity Time in Knowledge Graph. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, 1771–1776. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450356404.

Li, Y.; Sun, S.; and Zhao, J. 2022. TiRGN: Time-Guided Recurrent Graph Network with Local-Global Historical Patterns for Temporal Knowledge Graph Reasoning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2152–2158. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3.

Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; and Cheng, X. 2021. Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 408–417. ACM. ISBN 978-1-4503-8037-9.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2023. Learn from Relational Correlations and Periodic Events for Temporal Knowledge Graph Reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1559–1568. ACM. ISBN 978-1-4503-9408-6.

Lin, Q.; Liu, J.; Mao, R.; Xu, F.; and Cambria, E. 2023. TECHS: Temporal Logical Graph Networks for Explainable Extrapolation Reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1281–1293. Association for Computational Linguistics.

Liu, Y.; Ma, Y.; Hildebrandt, M.; Joblin, M.; and Tresp, V. 2022. TLogic: Temporal Logical Rules for Explainable Link Forecasting on Temporal Knowledge Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4120–4127.

Niu, G.; and Li, B. 2023. Logic and Commonsense-Guided Temporal Knowledge Graph Completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4569–4577.

Park, N.; Liu, F.; Mehta, P.; Cristofor, D.; Faloutsos, C.; and Dong, Y. ???? EvoKG: Jointly Modeling Event Time and Network Structure for Reasoning over Temporal Knowledge Graphs. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 794–803. ACM. ISBN 978-1-4503-9132-0.

Sun, H.; Zhong, J.; Ma, Y.; Han, Z.; and He, K. 2021. TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8306–8319. Association for Computational Linguistics.

Wang, X.; Sun, G.; Fang, X.; Yang, J.; and Wang, S. 2022. Modeling Spatio-temporal Neighbourhood for Personalized Point-of-interest Recommendation. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 3530–3536. ijcai.org.

Xiong, S.; Yang, Y.; Fekri, F.; and Kerce, J. C. 2023. TILP: Differentiable Learning of Temporal Logical Rules on Knowledge Graphs. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xu, D.; Ruan, C.; Körpeoglu, E.; Kumar, S.; and Achan, K. 2020. Inductive representation learning on temporal graphs.

In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xu, Y.; Ou, J.; Xu, H.; and Fu, L. 2023. Temporal Knowledge Graph Reasoning with Historical Contrastive Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4): 4765–4773.

Yu, X.; Chen, Z.; Ling, Y.; Dong, S.; Liu, Z.; and Lu, Y. 2023. Temporal Data Meets LLM – Explainable Financial Time Series Forecasting. arXiv:2306.11025.

Zhang, M.; Xia, Y.; Liu, Q.; Wu, S.; and Wang, L. 2023a. Learning Latent Relations for Temporal Knowledge Graph Reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12617–12631. Association for Computational Linguistics.

Zhang, M.; Xia, Y.; Liu, Q.; Wu, S.; and Wang, L. 2023b. Learning Long- and Short-term Representations for Temporal Knowledge Graph Reasoning. In Ding, Y.; Tang, J.; Sequeda, J. F.; Aroyo, L.; Castillo, C.; and Houben, G., eds., *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, 2412–2422. ACM.

Zhang, M.; Xia, Y.; Liu, Q.; Wu, S.; and Wang, L. 2023c. Learning Long- and Short-term Representations for Temporal Knowledge Graph Reasoning. In *Proceedings of the ACM Web Conference 2023*, 2412–2422. ACM. ISBN 978-1-4503-9416-1.

Zheng, S.; Yin, H.; Chen, T.; Nguyen, Q. V. H.; Chen, W.; and Zhao, L. 2023. DREAM: Adaptive Reinforcement Learning based on Attention Mechanism for Temporal Knowledge Graph Reasoning. In Chen, H.; Duh, W. E.; Huang, H.; Kato, M. P.; Mothe, J.; and Poblete, B., eds., *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, 1578–1588. ACM.