Expert Systems With Applications 286 (2025) 128081

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

FL-Evo: Jointly modeling fact and logic evolution patterns for temporal knowledge graph reasoning

Ruishen Liu [a], Xinzhi Wang [a], Shaorong Xie [a,b,*], Xiangfeng Luo [a,*], Huizhe Su [a]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China
[b] Engineering Research Center of Unmanned Intelligent Marine Equipment, Ministry of Education, Shanghai, 200444, China

ARTICLE INFO

Keywords:
Temporal knowledge graph
Large language model
Extrapolation reasoning
Unseen entity prediction

ABSTRACT

Temporal knowledge graphs (TKGs) extrapolation reasoning, intending to predict future events given the known KG sequence, benefits broad applications like policy-making and financial analysis. The key to this issue is to discern how knowledge evolves within these sequences. Currently, most works focus on modeling the evolution patterns through continuous sampling from TKGs, without ensuring the samples contain relevant facts or considering the knowledge beyond the samples. Faced with these challenges, we propose a novel model that performs prediction by capturing fact and logic knowledge evolution patterns (FL-Evo). For modeling fact evolution pattern, the fact knowledge is first distilled from large language models using designed prompts and subsequently refined with TKG. Then, entity-based subgraph sampling strategy extracts relevant facts from the TKG, capturing fact evolution patterns. Furthermore, logical knowledge mined from the TKG helps to derive the corresponding evolution pattern. Finally, the outputs of these two evolution patterns are integrated to realize the final prediction. Experimental results on five benchmark datasets demonstrate that FL-Evo outperforms existing temporal knowledge graph reasoning models, with improvements of up to 3.97 % in Hit@3 and 4.07 % in Hit@10. Notably, FL-Evo substantially enhances reasoning performance for unseen entities lacking prior records.

1. Introduction

Temporal Knowledge Graph Extrapolation Reasoning (TKGR), aimed at predicting future events given the known KG sequence, benefits various downstream applications, like financial analysis (Li & Sanna Passino, 2024) and policy-making (Deng, Rangwala, & Ning, 2020). In contrast to static knowledge graphs, which organize facts into triplets *(source, relation, target)*, temporal knowledge graphs expand this structure to quadruples, incorporating timestamps i.e., *(source, relation, target, timestamp)*. The quadruples indicate the facts are available at the specific timestamps. Due to the never-ending information and the limitations in knowledge extraction approaches, TKGs are incomplete in nature. Performing extrapolation reasoning based on incomplete facts is a formidable challenge.

Recently, many efforts have been devoted to learning the fact evolution patterns, such as the dynamic representation of entities and relations, to enhance TKGR performance. TTransE (Jiang et al., 2016) and TA-DisMult (García-Durán, Dumancic, & Niepert, 2018) complete TKG by adding temporal dimension into static knowledge graph reasoning approach. RE-NET (Jin, Qu, Jin, & Ren, 2019) employs RGCN to model the structure information of TKG. Reinforcement learning is used in TITer (Sun, Zhong, Ma, Han, & He, 2021), which constructs the temporal edges to solve the isolated temporal subgraphs problem. CyGNet (Zhu, Chen, Fan, Cheng, & Zhan, 2020) introduces copy-generation mechanisms to predict repetition facts. Tirgn (Li, Sun, & Zhao, 2022a) posits events that follow sequential, repetitive, and cyclical patterns, proposing a time-guided recurrent graph network for TKGR. THCN (Xu, Ou, Xu, & Fu, 2022) utilizes the causal convolutional network to realize prediction.

Although the above methods have advanced temporal knowledge extrapolation reasoning, there are still three challenges: (1) **Insufficient knowledge**. Current approaches rely on sampled TKGs, which neglects the knowledge beyond the samples (Fig. 1). This limitation results in inaccuracies in fact representation. (2) **Inefficient Sampling**. As shown in Fig. 1, related facts are not consistently present. Most previous methods sample temporal subgraphs continuously from TKG, without considering whether the samples contain the related facts, leading to inefficient evolution learning. (3) **Inability to Handle Unseen Entities.** Since entities are constantly emerging over time, effectively capturing evolving knowledge based on existing data becomes crucial. The limitation of

* Corresponding authors.
E-mail addresses: ruishenliu@shu.edu.cn (R. Liu), wxz2017@shu.edu.cn (X. Wang), srxie@shu.edu.cn (S. Xie), luoxf@shu.edu.cn (X. Luo), shzmerry@shu.edu.cn (H. Su).

https://doi.org/10.1016/j.eswa.2025.128081
Received 20 August 2024; Received in revised form 7 April 2025; Accepted 6 May 2025
Available online 15 May 2025
0957-4174/

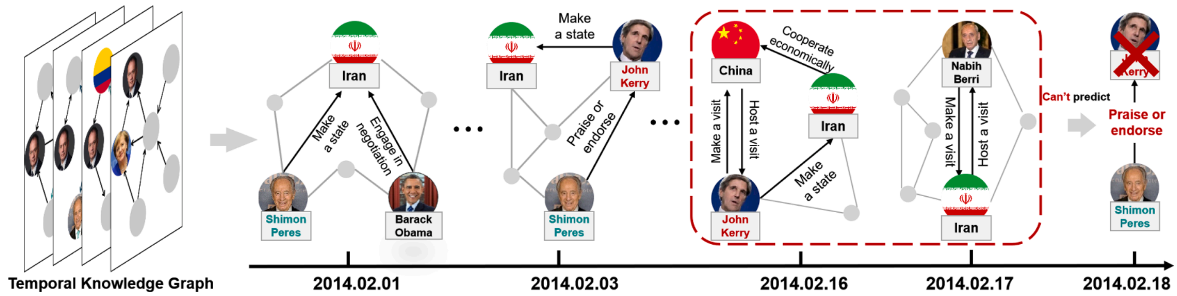


Fig. 1. The challenges of existing TKGR methods. These works only use the knowledge contained in the red rectangle. The green denotes the facts related to the query, and the relevant facts are out of the red rectangle. The red font is the correct answer.

insufficient information and inefficient sampling leads to a tough evolution acquisition, thereby degrading the reasoning performance on unseen entities.

To tackle the these issues, we propose the Fact and Logic Evolution model (FL-Evo), which leverages the knowledge from different aspects to capture various evolution patterns like fact and logic. FL-EVO consists of four components, including fact and logic knowledge (FLM), fact knowledge evolution pattern (FEM), logic knowledge evolution pattern (LEM) and fusion module (FM). The FLM serves as background knowledge base, incorporating fact and logic knowledge derived from LLM and TKG. For modeling the fact evolution pattern, the fact knowledge formed by entity and relation is drawn from a large language model (LLM) through various designed prompts and then refined with the TKG. Afterward, FEM employs the entity-based subgraph sampling strategy to extract the relevant subgraphs from TKG, capturing the fact evolution. In Fig. 1, given a query *(Shimon Peres, Praise or endorse, ?, 2014.02.18)*, the result *John Kerry* is realized via the relevant facts contained in temporal subgraphs which precede the target timestamp *2014.02.18*, i.e., *2014.02.01* and *2014.02.03*. For the logic evolution pattern, LEM first extracts the logic knowledge like temporal rules from TKG. The temporal rules are then utilized to derive the logic knowledge evolution, assisting in reasoning. The prediction *John Kerry* can be obtained via the temporal rule *praise or endorse* ← *praise or endorse*. FM realizes the final prediction *John Kerry* by fusing the outputs from these evolution patterns like fact and logic knowledge evolution patterns. The proposed model leverages known knowledge from different aspects to explore knowledge evolution patterns, boosting reasoning performance. Overall, the contributions of the proposed model are as follows:

- We propose a novel model for temporal knowledge graph extrapolation reasoning that leverages existing knowledge from diverse perspectives to model evolution patterns like fact and logic, enhancing reasoning performance.
- Various prompts are designed to extract world fact knowledge from large language models, enhancing model generalization and enabling FL-Evo to handle unseen entities.
- We conduct extensive experiments on five benchmark datasets, the results outperform baseline methods and significantly improve MRR and Hit@10, demonstrating improvements of up to 3.97 % and 4.07 % in Hit@3 and Hit@10. Especially, reasoning performance on unseen entities indicates the effectiveness of the proposed method.

The remaining paper is structured as follows. Some related works are introduced in Section 2. In Section 3, we first define the notations used in this paper and then elaborate on the proposed model in detail. Section 4 reports the experimental setup and the results. Finally, conclusions are drawn in Section 5.

2. Related work

In this section, we first distinguish the difference between static knowledge graph reasoning and temporal knowledge graph reasoning. Then, we introduce the related work over temporal KG reasoning from interpolation and extrapolation settings.

2.1. Reasoning method for static knowledge graph

Static KG reasoning, predicting the missing facts from existing information, can be divided into two categories, i.e., embedding-based model and multi-hop model. Embedding-based models infer the missing facts by mapping entities and relations into multi-dimensional vector space. There are bilinear models (Trouillon, Welbl, Riedel, Gaussier, & Bouchard, 2016), distance-based models (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013), and neural network models (Dettmers, Minervini, Stenetorp, & Riedel, 2018). Multi-hop reasoning models achieve the target entities through traveling on the knowledge graph edges. DeepPath (Xiong, Hoang, & Wang, 2017) is the first to adopt reinforcement learning to perform multi-hop reasoning. RLH (Wan, Pan, Gong, Zhou, & Haffari, 2020) leverages hierarchical reinforcement learning to predict missing facts. AnyBURL (Meilicke, Chekol, Ruffinelli, & Stuckenschmidt, 2019) extracts horn rules and employs them for reasoning tasks. However, these methods are not suitable for temporal KG reasoning due to their disregard for temporal information.

2.2. Reasoning method for temporal knowledge graph

Temporal KG Reasoning consists of interpolation and extrapolation (Jin et al., 2019). Interpolation reasoning infers the missing facts within the existing timestamp. TA-TransE (García-Durán et al., 2018) and TTransE (Jiang et al., 2016) conduct temporal reasoning by adding a temporal dimension to the static reasoning method, which injects temporal labels into each relation. TA-DisMult (García-Durán et al., 2018) decomposes the timestamp into the sequence of temporal tokens. However, interpolation methods are limited in predicting future events due to their reliance on known timestamps. Extrapolation reasoning infers events at future timestamps by utilizing the existing information. CluSTeR (Zhu et al., 2020) employs reinforcement learning to extract relevant facts. RE-GCN (Li et al., 2021) enhances reasoning performance by integrating entity attributes from static KGs into temporal KGs. TLogic (Liu, Ma, Hildebrandt, Joblin, & Tresp, 2022) extracts the temporal rule from the TKG and achieves target entities based on the high-confidence rules. The transformer is used in rGalT (Gao et al., 2022) to realize the temporal knowledge extrapolation reasoning. LMS (Zhang, Hui, Mu, & Tian, 2024) introduces a novel method to perform temporal knowledge graph reasoning by using multi-graph learning. Dynamic hypergraph embedding is used in DHE-TKG (Liu et al., 2024). KGTransformer (Li & Sanna Passino, 2024) introduces an attention-based GNN to conduct reasoning on financial dataset. THCN (Chen et al., 2024) employs causal convolutional network to realize prediction. TaReT (Ma et al., 2024) integrates topological relation graphs and temporal fusion information to perform reasoning.

Existing works sample subgraphs continuously from TKG to perform prediction, without ensuring the inclusion of relevant facts. This

Table 1
Notation Table.

Symbol	Meaning	Symbol	Meaning
$\mathcal{E}$	Enity set	$\mathcal{R}$	Relation Set
$\mathcal{G}$	Temporal Knowledge graph	$\mathcal{G}^{t_i}$	ith temporal subgraph
$\lvert\mathcal{E}\rvert$	The number of entities	$\lvert\mathcal{R}\rvert$	the number of relations
e_s	Source entity	r_q	Query relation
t_q	Query timestamp	e_o	Target entity
$\mathcal{G}_{e_s}^{t_i}$	the ith temporal subgraph containing e_s	t_i	the timestamp of ith temporal subgraph
m	the number of sampled subgraphs	$\mathcal{G}_s$	Static Knowledge Graph
H	Embedding	$softmax(\cdot)$	softmax function
LLM	Large language model	FLM	Fact and Logic Knowledge Module
FEM	Fact Knowledge Evolution Module	LEM	Logic Knowledge Evolution Module

strategy is inadequate to capture evolution patterns, leading to degraded performance in predicting unseen entities lacking records. The proposed model utilizes information from various perspectives to model knowledge evolution, enhancing TKG extrapolation reasoning.

3. Method

In this section, we first introduce the task of temporal knowledge graph extrapolation reasoning and some notations (Table 1). Then the overview framework of FL-Evo and the details are given. Finally, we describe the loss function leveraged in the training process.

3.1. Notations

Temporal KG. TKG is regarded as a sequence of subgraphs, ordered by timestamp, i.e., $\mathcal{G} = \{\mathcal{G}^{t_0}, \mathcal{G}^{t_1}, \dots, \mathcal{G}^{t_T}\}$, where $\mathcal{G}^{t_m} = (\mathcal{E}, \mathcal{R}, \mathcal{F}^{t_m})$. $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations across the entire timestamp, while $\mathcal{F}^{t_m}$ is the facts at time t_m i.e., (e, r, o, t_m). The quadruple represents there is a relation r $\in \mathcal{R}$ between entities $e, o \in \mathcal{E}$ at timestamp t_m. The inverse relation is often added into the dataset, i.e., (o, r^{-1}, e, t_m). Meanwhile, the static KG can be treated as $\mathcal{G}_s = (\mathcal{E}_s, \mathcal{R}_s, \mathcal{F}_s)$, where represent entity, relation, and fact set respectively.

Temporal KG Extrapolation Reasoning. Temporal Knowledge Graph Extrapolation Reasoning aims to predict events at timestamp t_q using the facts occurring before t_q, i.e., $(e_s, r_q, ?, t_q)$ or $(?, r_q, e_o, t_q)$. For each prediction at timestamp t_q, the history temporal subgraphs can be represented as $\mathcal{G}_{e_s}^{t_0:t_{q-1}}$. $\mathcal{G}_{e_s}^{t_i}$ denotes the ith temporal subgraph, containing the entity e_s. $\mathcal{G}_{e_s}^{t_{q-1}}$ indicates the temporal subgraph containing the entity e_s, which is closest to the query timestamp, and t_{q-1} is the corresponding timestamp.

Temporal Rule. Temporal rule with confidence mined from TKG can be formulated as the conjunction of atoms with timestamps

$$conf \quad r_q(e_s, e_o, t_q) \leftarrow r_1(e_s, z_1, t_1), \dots, r_n(z_{n-1}, e_o, t_n) \tag{1}$$

where $t_1, \dots, t_n < t_q$. r_1 and r_n are relation variants, while z_1 and z_{n-1} are entity variants. t_1 and t_n are timestamps. $conf$ is the confidence score of the temporal rule. For the prediction $(e_s, r_q, ?, t_q)$, the candidate set is regard as $C_{e_s}^{rule}$, obtained through temporal rule. In this paper, $r_q(e_s, e_o, t_q)$ is equal to (e_s, r_q, e_o, t_q). To simplify the expression of the temporal rule, $r_q(e_s, e_o, t_q) \leftarrow r_1(e_s, z_1, t_1), \dots, r_n(z_{n-1}, e_o, t_n)$ is equal to $r_q \leftarrow r_1, \dots, r_n$.

3.2. Model overview

The framework of FL-Evo is shown in Fig. 2. FL-Evo utilizes the existing knowledge from different aspects to capture diverse evolutionary patterns including fact and logic. The proposed model includes fact and logic knowledge (FLM), fact knowledge evolution pattern (FEM), logic knowledge evolution pattern (LEM) and fusion module (FM). The FLM is treated as background knowledge base, incorporating fact and logic knowledge derived from LLMs and TKG. For the fact knowledge evolution pattern, the fact knowledge formed by entity and relation is first drawn from LLMs through designed prompts, followed by refined

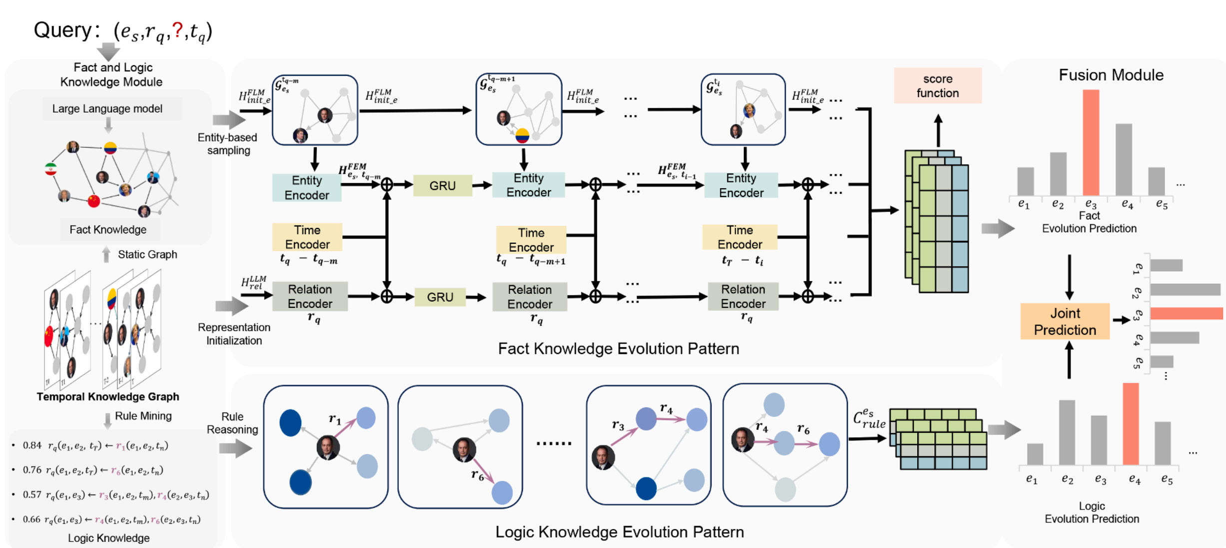


Fig. 2. The FL-Evo consists of fact and logic knowledge (FLM), fact knowledge evolution pattern(FEM), logic knowledge evolution pattern(LEM) and fusion module (FM). The FLM, integrated with LLM, stores the entire temporal knowledge graph and world knowledge, initializing the fact representation and mining the temporal rules. FEM selects temporal subgraphs containing the relevant facts, effectively learning the fact evolution. Temporal rules are used to obtain the logic evolution in LEM. The final prediction is achieved by fusing the output of these evolution patterns. m represents the number of sampled subgraphs. The pictures are the source entity e_s. $\mathcal{G}_{e_s}^{t_{q-m}}$ represents the mth sampled subgraph containing e_s, and t_{q-m} denotes the timestamp of the corresponding subgraph.

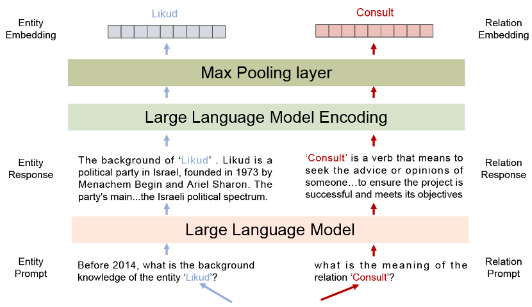


Fig. 3. The example of the Large Language Model and the prompts used in the proposed model.

with TKG. FEM then adopts the entity-based subgraphs sampling strategy to extract the relevant subgraphs from the TKG, aiding in capturing the fact evolution patterns. For logic evolution modeling, LEM extracts logical knowledge, such as rules, from the TKG. These rules are then used in LEM to model the corresponding evolution patterns. Finally, FM integrates outputs from these evolution patterns to predict future events.

3.3. Initialize fact knowledge representation

To enhance reasoning accuracy, it is crucial to leverage the existing knowledge. Previous works only utilize knowledge contained in the TKGs, ignoring a lot of valuable knowledge. In the proposed model, the FLM incorporates not only the entire TKG but also world knowledge.

Entity and Relation Prompts. As the never-ending information and the limitations of knowledge extraction methods, the existing TKGs contain finite facts, leading to deficiencies in entities and relations representation. Leveraging the knowledge stored in LLMs, which are trained on extensive data, enriches the background knowledge of TKG. However, LLMs also contain biases that could affect the quality of the generated facts. To address this issue, we employ specific entity and relation prompts to guide LLMs in generating facts.

Furthermore, to ensure the integrity of temporal knowledge extraction and minimize biases in the responses, temporal prompts are constructed. These temporal prompts are restricted to timestamps and entities, without additional information, minimizing the influence of irrelevant context. As illustrated in Fig. 3, the ICEWS14[1] dataset records facts that occurred during 2014. For the entity *Likud*, the prompt can be constructed as '*Before 2014, what's the background knowledge of the entity Likud?*'.

Compared to entities, as the relations meaning remains relatively stable, the relation prompt omits the temporal information. For example, for the relation '*Consult*', the relation prompt can be constructed as '*what's the meaning of the relation Consult*'.

Given the prompts, the LLM feedbacks relevant descriptions. Then, the descriptions are encoded with LLM (Fig. 3). Finally, a pooling layer is used to reconstruct the description embedding i.e., H^{LLM}_{ent} and H^{LLM}_{rel}:

$$H^{LLM}_{ent} = MaxPooling(LLM(prompt_{ent})) \tag{2}$$

$$H^{LLM}_{rel} = MaxPooling(LLM(prompt_{rel})) \tag{3}$$

$prompt_{ent}$ and $prompt_{rel}$ denote entities and relations prompts respectively. $Maxpooling(\cdot)$ is a pooling layer and LLM is a large language model. $H^{LLM}_{ent} \in \mathbb{R}^{|\mathcal{E}|\times d_1}$ and $H^{LLM}_{rel} \in \mathbb{R}^{2|\mathcal{R}|\times d_2}$ can be regarded as the initial representation of entities and relations. d_1 and d_2 denote the dimension of entity and relation embeddings. The details of prompts are illustrated in Appendix A.

[1] The integrated crisis early warning system

Static Knowledge Graph Module. Most previous TKGR methods only utilize information within a limited set of sampled temporal subgraphs. However, the perception of the entities requires consideration across their entire temporal history up to the query timestamp. Additionally, the initial embeddings for entities and relations, generated by LLMs, may contain biases that negatively impact reasoning performance. To address these issues, we refine the embeddings by leveraging a static knowledge graph, which helps mitigate the biases and improve the quality of the representations. Given a query $(e_s, r_q, ?, t_q)$, we convert all temporal subgraphs preceding t_q into static knowledge graph $\mathcal{G}^{t_q}_s$ by disregarding the timestamps. The facts within $\mathcal{G}^{t_q}_s$ can be aggregated through RGCN[2], yielding the comprehensive representations of entities.

$$H^{(l+1)}_{init\_e} = f\Big(\sum_{(e,r,o)\in\mathcal{G}^{t_q}_s} \Big(\frac{W^l_r H^l_o}{|\mathcal{N}_e|}\Big) + W^l_{loop} H^l_e \Big) \tag{4}$$

W_r and W_{loop} is the learnable parameters. $|\mathcal{N}_e|$ is determined by the number of neighboring entities for a given entity e. H^0_e is H^{LLM}_{ent}, while $H^{(l+1)}_{init\_e}$ is regarded as $H^{FLM}_{init\_e} \in \mathbb{R}^{|\mathcal{E}|\times d_1}$.

3.4. Fact knowledge evolution pattern modeling

As illustrated in Fig. 1, most previous works learn the fact evolution patterns within finite sampled temporal subgraphs. Therefore, it is essential to incorporate temporal subgraphs related to the query. In FEM, we first sample the relevant temporal subgraphs and then capture the fact evolution pattern (Fig. 4).

Entity-based Subgraph Sampling. The existing methods adopt continuous subgraph sampling, without ensuring the sampled subgraphs encompass the relevant facts, which fails to capture the fact evolution (Fig. 1). Given a query $(e_s, r_q, ?, t_q)$, we first extract all subgraphs containing the source entities e_s i.e., $\mathcal{G}^{t_0:t_{q-1}}_{e_s}$. Finite subgraphs are then sampled from the extracted temporal subgraphs to learn evolution patterns. The closer a sampled subgraph is to the query timestamp, the greater its impact on reasoning performance (Han, Chen, Ma, & Tresp, 2021). Thereby, the exponential distribution is utilized for temporal subgraph sampling. As illustrated in Fig. 4, given a query *(Likud, Consult, ?, 2014.11.12)*, the temporal subgraphs preceding *2014.11.12* are obtained, i.e., *2014.09.10*, *2014.09.16* and *2014.10.10*.

Time Encoder. Different from the previous works sampling continuously, FL-Evo extracts the entity-based temporal subgraphs, leading to the sparsity of time interval, i.e., $\Delta t = t_q - t_{\hat{i}}$, where $t_{\hat{i}}$ is the timestamp of the sampled subgraph. As the occurrence of the facts follows cyclical patterns (Li et al., 2022b), the cosine function is used to encode time Eq. (5):

$$H_{\hat{i}} = cos(W_{\hat{i}}\Delta t + b_{\hat{i}}) \tag{5}$$

$cos(\cdot)$ is the cosine function, $W_{\hat{i}}$ is a learnable parameter and $b_{\hat{i}}$ denotes bias.

Entity Encoder. The structure representation of TKG captures the correlation between entities and relations. Even though the GCN can effectively aggregate the graph structure information, it tends to overshadow the initial meaning of the entity through multi-hop aggregation (Zhao, Zhang, Kong, & Yin, 2021). To address this issue, we augment the final aggregation by incorporating the initial entity representation.

$$h^{loop}_{e,t} = W_{loop} h^0_{e,t} \tag{6}$$

$$agg^{l+1}_{e,t} = \frac{1}{|\mathcal{N}_e|} \sum_{(e,r,o)\in\mathcal{G}^t_{e_s}} \left(W_e h^l_{o,t} + W_r h_{r,t} \right) \tag{7}$$

$$h^{l+1}_{e,t} = h^{loop}_{e,t} + agg^{l+1}_{e,t} \tag{8}$$

[2] https://github.com/JinheonBaek/RGCN

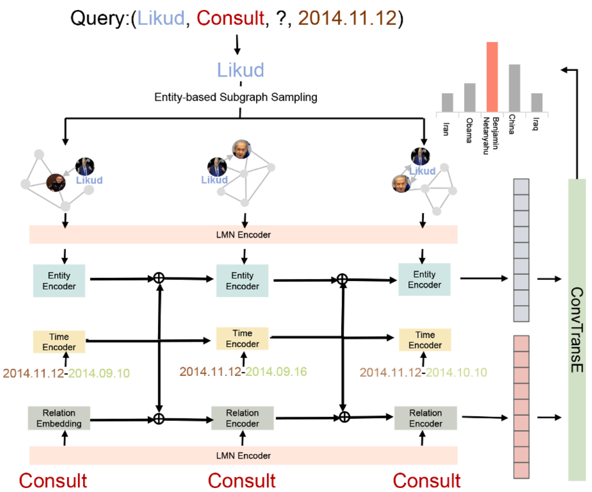


Fig. 4. The reasoning process of temporal representation evolution in FEM. We extract relevant facts using entity-based subgraph sampling strategy. The blue is the facts related to the query. The green denotes the timestamp of the subgraph containing the source entity.

W_{loop}, W_e and W_r are the learnable parameters, and $h^l_{o,t}$ is the lth layer embeddings of entity o at timestamp t. $h_{r,t}$ is the relations embeddings at timestamp t. $h^0_{e,t}$ is $H^{FLM}_{init\_e}$ and $h^{l+1}_{e,t}$ is regarded as $H^{FEM}_{e,t}$.

In the TKG, the entities meaning evolves over time. We integrate entity embedding with time encoding to depict the dynamic nature of entities, i.e., $H_{e,t} = [H^{FEM}_{e,t} \,||\, H_t]$, where $H^{FEM}_{e,t}$ is the embedding of e at timestamp t, H_t denotes the timestamp t embedding, and || signifies the concatenation operation. GRU is used to learn the entity evolution pattern:

$$H'_{e,t} = GRU(H'_{e,t-1}, H_{e,t}) \tag{9}$$

$H'_{e,t-1}$ and $H'_{e,t}$ are the hidden embedding of entity e at t-1 and t.

Relation Encoder. Similarly, the relation embedding contains not only the static relation representation but also the time embedding, i.e., $H_{r,t} = [H^{LLM}_{rel} || H_t]$, where H^{LLM}_{rel} denotes the initial relation representation, obtained from LLM (Eq. (3)), H_t is the timestamp embedding, and || signifies the concatenation operation. GRU is also utilized to obtain the evolution pattern through the timeline:

$$H'_{r,t} = GRU(H'_{r,t-1}, H_{r,t}) \tag{10}$$

$H'_{r,t-1}$ and $H'_{r,t}$ are the hidden embedding of relation r at t-1 and t timestamps.

3.5. Logical knowledge evolution pattern modeling

Fact evolution modeling is inclined to reasoning the events with repetition (Li et al., 2022a), failing to deal with unseen entities. To address this issue, previous research treats all entity vocabulary as candidate sets, assuming candidate entities follow a uniform distribution. However, entities should have different probabilities relying on the context. For example, as shown in Fig. 1, if the path *Praise and endorse* holds, the result of *Praise and endorse* is more likely to be an entity related to the path.

In the FL-Evo, LEM captures logic evolution based on temporal rules extracted from the entire TKG. FL-Evo employs temporal rule mining methods, like TLogic (Liu et al., 2022) and StreamLearner (Omran, Wang, & Wang, 2019), to derive rules automatically from the TKG. As the same head holds different bodies like $rule_1$: *0.86 Praise and endorse ← Praise and endorse* and $rule_2$: *0.54 Praise and endorse ← Discuss by telephone, Make an appeal or request, Engage in negotiation.* Given a query $(e_s, r_q, ?, t_q)$, each rule generates its own rule candidate set,

$$C^{rule}_{e_s,r_q} = C^{rule_1}_{e_s,r_q} \cup C^{rule_2}_{e_s,r_q} \ldots C^{rule_n}_{e_s,r_q} \tag{11}$$

where $C^{rule_i}_{e_s,r_q}$ is a candidate entity set based on the ith rule. Since different rules hold different confidence, candidates derived from higher-confidence rules should receive higher scores (Liu et al., 2022). In addition, candidates supported by multiple rules should be assigned higher scores:

$$Score(e) = 1 - \Pi_{conf(e)|e\in C^{rule}_{e_s,r_q}} (1 - conf(e)) \tag{12}$$

where $conf(e)$ is the confidence of the rule generating the candidate entity e. Given a query $(e_s, r_q, ?, t_q)$, if candidate entities exist in $C^{rule}_{e_s,r_q}$, the $\mathcal{M}^{rule}_{e_s,r_q}$ obtains corresponding score. Otherwise, the score is 0. $\mathcal{M}^{rule}_{e_s,r_q}$ is an N-dimensional vector. N is equal to the number of entities $|\mathcal{E}|$.

3.6. Information fusion modeling

The fusion module integrates the outputs from different evolution patterns to predict future events.

Representation evolution decoder. The entity and relation embedding can be obtained from FEM. Li et al. (2021) regards a convolutional neural network as a decoder, achieving better results. In this paper, ConvTransE (Zhen, Wang, Zhou, Fang, & Quan, 2018)[3] is regarded as the decoder:

$$\mathbf{P}^{FEM} = ConvTransE(H_{e,t}, H_{r,t}) \tag{13}$$

$H_{e,t}$ and $H_{r,t}$ is the embedding of entity e and relation r at the timestamp t, obtained from Eqs. (9) and (10).

$$ConvTransE(H_{e,t}, H_{r,t}) = f(vec(M(H_{e,t}, H_{r,t}))W) \tag{14}$$

$M(H_{e,t}, H_{r,t})$ is the output from the convolution layer. $vec(\cdot)$ transforms the feature matrix into a vector. $f(\cdot)$ signifies the activation function and W is a learnable parameter.

Logic evolution decoder. From LEM, the $\mathcal{M}^{rule}_{e_s,r_q}$ denotes the candidate set based on temporal rules. Softmax is used to decode the $\mathcal{M}^{rule}_{e_s,r_q}$:

$$\mathbf{P}^{LEM} = softmax(\mathcal{M}^{rule}_{e_s,r_q}) \tag{15}$$

Scoring Function. After decoding various evolution patterns, we use weight factor α to perform the final predictions.

$$\mathbf{P}^{final} = (1-\alpha)\mathbf{P}^{FEM} + \alpha\mathbf{P}^{LEM} \tag{16}$$

3.7. Training objective

Given a query $(e_s, r_q, ?, t_q)$, entity prediction can be seen as a multi-label learning problem. We employ cross-entropy during extrapolation reasoning:

$$C^e = \sum_{(e_s,r_q,e_o,t_q)\in\mathcal{G}^{t_q}} y^e_{t_q} log\mathbf{P}(e_o|e_s, r_q, t_q) \tag{17}$$

where $y^e_{t_q}$ is equal to 1, if $e = e_o$, otherwise 0. $\mathbf{P}(e_o|e_s, r_q, t_q)$ is the final probability of entity prediction.

4. Experiments

This section elucidates the effectiveness of FL-Evo on temporal knowledge graph reasoning. We first describe the details of the experimental setting, like baseline methods and evaluation metrics. Secondly, we present comparative results. Then the results on unseen and seen entities are discussed. After that, we analyze the importance of hyperparameters, followed by the ablation study. Finally, case study is provided to illustrate the reasoning process of our model.

[3] https://github.com/Lee-zix/RE-GCN

Dataset	$\vert\mathcal{E}\vert$	$\vert\mathcal{R}\vert$	Train	Valid	Test
ICEWS14	7,128	230	63,685	13,823	13,222
ICEWS0515	10,488	251	368,868	46,302	46,159
ICEWS18	23,033	256	373,018	45,995	49,545
GDELT	7,691	240	1,734,399	238,765	305,241
FinDKG	13,645	15	119,549	11,444	13,069

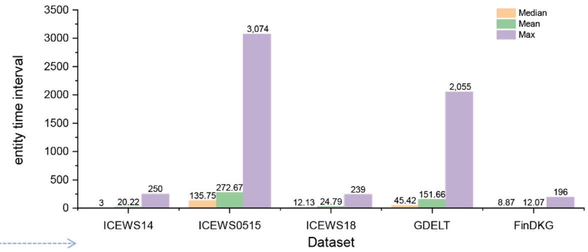


Fig. 5. The details of dataset. $|\mathcal{E}|$ and $|\mathcal{R}|$ denote the number of entities and relations respectively. Mean, Median and Max signify the average, median, and max of the entity temporal interval.

4.1. Experiment setup

Dataset. Five datasets are used to evaluate the proposed model i.e., ICEWS14, ICEWS05-15, ICEWS18 (Boschee et al., 2015), GDELT (Tone, 2015) and FinDKG (Li & Sanna Passino, 2024). The first three datasets are from the Integrated Conflict Early Warning System (ICEWS), containing political events, finance events and so on. GDELT is sampled from the global database of events, language, and tone. FinDKG is developed from financial news articles collected from the Wall Street Journal spanning from 1999 to 2023. All of these datasets are event-based, and each event is time-stamped. Given the task of extrapolation reasoning, the dataset is split into the train, valid, and test (timestamps of train $<$ timestamps of valid $<$ timestamp of test). The details of dataset are shown in Fig. 5, including entities temporal intervals.

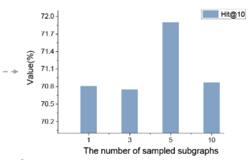
Baseline and Evaluation Metrics. Three types of reasoning models are treated as baselines: (1) static knowledge graph reasoning: ComplEx (Trouillon et al., 2016), R-GCN (Schlichtkrull et al., 2017), and ConvTransE (Zhen et al., 2018); (2) Interpolation reasoning: TA-Dismult (García-Durán et al., 2018), TNTCompIEX (Lacroix, Obozinski, & Usunier, 2020) and DE-SimpIE (Goel, Kazemi, Brubaker, & Poupart, 2020); (3) Extrapolation reasoning: TITer (Sun et al., 2021), RE-GCN (Li et al., 2021), CluSTeR (Zhu et al., 2020), RE-Net (Jin et al., 2019), xERTE (Han et al., 2021), TLogic (Liu et al., 2022), GHT (Sun, Geng, Zhong, Hu, & He, 2022), rGalT (Gao et al., 2022), PPT (Xu, Liu, Peng, Jia, & Peng, 2023), HGLS (Zhang, Xia, Liu, Wu, & Wang, 2023), TECHS (Lin, Liu, Mao, Xu, & Cambria, 2023), TR-Rules (Li et al., 2023), THCN (Chen et al., 2024), KGTransformer (Li & Sanna Passino, 2024) and TaReT (Ma et al., 2024). The mean reciprocal rank (MRR) and Hit@{1, 3, 10} are used as evaluation metrics.

The details of implementation. As temporal rules play an important role in the LEM, we leverage TLogic (Liu et al., 2022)[4] to mine temporal rules from temporal knowledge graphs. LLAMAv2-13B [5] is used as LLM (Touvron et al., 2023). The embedding dimension of entities and relations is 200. We use Adam (Kingma & Ba, 2014) as an optimizer and the learning rate is 0.001. The number of sampled subgraphs and factor α are chosen from {1, 3, 5, 10} and $\alpha \in (0.0, 1.0)$. The epoch is set to 15. The code and data of FL-EVO is available at https://github.com/ruishenliu/FL-EVO.

4.2. Results on TKG

The results on MRR and Hit@{1, 3, 10} are elaborated in Tables 2 and 3. Compared to baseline models, FL-Evo consistently improves Hit@3 by 3.97 %, 0.16 %, 0.21 %, 0.63 % and 19.47 %, and shows significant enhancements in Hit@10 by 1.49 %, 4.07 %, 0.34 %, 0.16 % and 27.39 %. The FL-EVO also achieves significant improvement in the financial dataset. As static reasoning methods disregard timestamp information, the results of the static reasoning models are almost the worst. FL-Evo outperforms interpolation reasoning models because interpolation reasoning relies on known event timestamps, limiting their ability to predict future events efficiently. The proposed model also surpasses most extrapolation reasoning models. Despite RE-GCN utilizing global and local information to capture the event evolution pattern, it adopts continuous sampling, limiting their ability to represent facts. CluSTeR and TITer extract subgraphs using reinforcement learning, without considering the relevant facts. The entity-based subgraph sampling strategy ensures each extracted subgraph contains the relevant facts, enabling the learning of fact evolution patterns. By integrating LLMs into FLM, FL-Evo obtains more information and enriches the fact knowledge formed by entity and relation. TLogic searches the related history information through rules. However, due to limited sampled subgraphs, they are unable to make full use of the existing information. As proposed in FL-Evo, sufficient information is leveraged from different aspects to explore evolution patterns.

Number	MRR	Hit@1	Hit@3	Hit@10
1	47.08	34.93	53.25	70.81
3	47.08	34.99	53.07	70.75
5	48.09	35.63	55.00	71.90
10	47.24	35.00	53.56	70.87



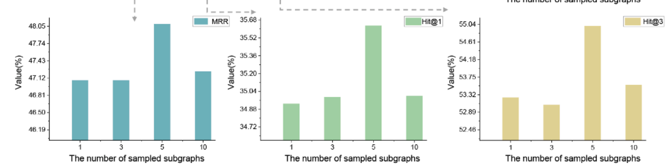


Fig. 6. The sensitivity analysis of the number of the sampled subgraphs on ICEWS14.

4.3. Unseen and seen entity analysis

As the never-ending information, unseen entities continually emerge. In the ICEWS14 and ICEWS18 datasets, there are 25.79 % and 33.38 % entities without history, respectively. Previous work often suffers from inadequate information and inefficient sampling, leading to inferior reasoning performance on unseen entities. To tackle with this issue, FL-Evo makes use of the knowledge from various aspects to model evolution patterns. We adopts the FLM integrated with LLMs, providing additional information. The FEM captures fact evolution pattern through entity-based subgraphs sampling strategy. The learned evolution patterns can be extended to unseen entity prediction. Additionally, LEM explores logic evolution to assist in reasoning.

We conduct experiments on both seen and unseen entities using dedicated test datasets based on ICEWS14 and ICEWS18. The test datasets are divided into two parts, containing only seen and unseen entity queries, respectively (Appendix B). Tables 4 and 5 report that the proposed model consistently outperforms others. Although TLogic leverages temporal rules for TKGR, its performance is hindered by limitations in the number of sampled subgraphs. FEM and FLM make significant

[4] https://github.com/liu-yushan/TLogic
[5] https://huggingface.co/meta-llama

Table 2
The entity prediction results on ICEWS14, ICEWS0515 and ICEWS18. All results are percentages. The best results are in bold.

	ICEWS14				ICEWS0515				ICEWS18			
Model	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
ComplEx	30.84	21.51	34.48	49.59	31.69	21.44	35.74	52.04	21.01	11.87	23.47	39.94
R-GCN	28.03	19.42	31.95	44.83	27.13	18.83	30.41	43.16	15.05	8.13	16.49	29.00
ConvTransE	31.50	22.46	34.98	50.03	30.28	20.79	33.80	49.95	23.22	14.26	26.13	41.34
TA-DistMult	26.22	16.83	29.72	45.23	27.51	17.57	31.46	47.32	16.42	8.60	18.13	32.51
TNTCompIEX	34.05	25.08	38.50	50.92	27.54	9.25	30.80	42.86	21.23	13.28	24.02	36.91
DE-SimpIE	33.36	24.85	37.15	49.82	35.02	25.91	38.99	52.75	19.30	11.53	21.86	34.80
TITer	36.06	27.51	40.16	52.05	38.11	26.83	44.43	59.44	25.34	18.09	28.17	38.95
RE-GCN	37.78	27.17	42.50	58.84	38.27	27.43	43.06	59.93	27.51	17.82	31.17	46.55
CluSTeR	46.0	33.8	–	71.2	44.6	34.9	–	63.0	32.3	20.6	–	55.9
RE-NET	35.77	25.99	40.10	54.87	36.86	26.24	41.85	57.60	26.17	16.43	29.89	44.37
xERTE	40.79	32.70	45.56	57.30	46.62	37.84	52.31	63.92	29.31	21.03	33.51	46.48
TLogic	43.04	33.56	48.27	61.23	46.97	36.21	53.13	67.43	29.82	20.54	33.95	48.53
GHT	37.40	27.77	41.66	56.19	41.5	30.79	46.85	67.73	27.40	18.08	30.76	45.76
rGalT	38.33	28.57	42.86	58.13	38.89	27.58	44.19	59.10	27.88	18.01	31.59	47.02
PPT	38.42	28.94	42.5	57.01	38.85	28.57	43.35	58.63	26.63	16.94	30.64	45.43
HGLS	47.00	35.06	–	70.41	46.21	35.32	–	67.12	29.32	19.21	–	49.83
TECHS	43.88	34.59	49.36	61.95	48.38	38.34	54.69	68.92	30.85	21.81	35.39	49.82
TR-Rules	43.32	33.96	48.55	61.17	47.64	37.06	53.80	67.57	30.41	21.10	34.58	48.92
THCN	45.39	36.5	50.84	66.07	51.94	40.32	57.79	72.18	35.63	24.90	39.26	56.76
TaReT	47.56	**36.04**	51.03	69.32	52.39	39.23	58.69	72.63	34.98	**24.68**	39.41	56.76
KGTransformer	23.98	–	26.89	41.22	–	–	–	–	–	–	–	–
FL-Evo	**48.09**	35.63	**55.00**	**71.90**	**52.48**	**40.05**	**58.85**	**76.70**	**35.43**	24.47	**39.62**	**57.10**
	+0.53	–	+3.97	+1.49	+0.09	+0.82	+0.16	+4.07	+0.45	–	+0.21	+0.34

Table 3
The entity prediction results on GDELT and FinDKG. All results are percentages. The best results are in bold.

	GDELT				FinDKG			
Model	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
ComplEx	9.84	5.17	9.58	18.23	–	–	–	–
R-GCN	12.17	7.40	12.37	20.63	6.17	–	6.87	10.13
ConvTransE	19.07	11.85	20.32	33.14	–	–	–	–
TA-DistMult	11.17	5.09	11.58	22.65	–	–	–	–
TNTCompIEX	–	–	–	–	–	–	–	–
DE-SimpIE	–	–	–	–	–	–	–	–
TITer	15.75	10.94	15.74	25.37	–	–	–	–
RE-GCN	19.31	11.99	20.61	33.59	–	–	–	–
RE-NET	26.17	16.43	29.89	44.37	10.95	–	11.89	18.17
CluSTeR	18.3	11.6	-	31.9	–	–	–	–
xERTE	19.45	11.92	20.84	34.18	–	–	–	–
TLogic	21.07	13.39	23.24	37.07	–	–	–	–
GHT	20.04	12.68	21.37	34.42	–	–	–	–
rGalT	19.56	12.11	20.89	34.15	–	–	–	–
PPT	–	–	–	–	–	–	–	–
HGLS	19.04	11.79	–	33.23	–	–	–	–
TECHS	–	–	–	–	–	–	–	–
TR-Rules	–	–	–	–	–	–	–	–
THCN	23.46	15.18	25.21	39.03	–	–	–	–
TaReT	23.03	15.26	24.63	39.42	–	–	–	–
KGTransformer	—	–	–	–	12.45	–	13.76	21.13
FL-Evo	**24.01**	**16.04**	**25.26**	**39.58**	**30.38**	**21.19**	**33.23**	**48.52**
	+0.98	+0.78	+0.63	+0.16	+17.93	–	+19.47	+27.39

contributions to the results, thereby validating the effectiveness of the proposed model in leveraging existing information.

4.4. Sensitivity analysis

The number of the sampled temporal subgraphs. The learning of the evolution is determined by the information within the finite sampled subgraphs. Thereby, the reasoning performance is sensitive to its size. Fig. 6 presents that the optimal result is achieved when it is equal to 5. If the size is less than 5, there is insufficient information to support the pattern learning. Conversely, since the more recent facts have a greater impact on inference, the introduction of more premature facts affects inference performance.

Factor Weight α. LEM introduces temporal rules to assist reasoning. The weight factor α is used in FM to balance the importance of FEM and LEM. As shown in Fig. 7, when the α is equal to 0.5, the reasoning performance peaks. It indicates both representation and logic evolution patterns are important.

4.5. Entity embedding analysis

In FL-EVO, we initialize entity and relation embedding using a LLM through various prompts and subsequently refine the representation with TKG. To verify the ability of the proposed encoding methods, Fig. 8 illustrates a 2D visualization of entity embedding on ICEWS14. Each node denotes an entity embedding, and colors denote different groups,

Table 4
The unseen entity prediction results on ICEWS14 and ICEWS18. All results are percentages. We rerun their codes with the released hyperparameters. The best results are in bold. "w/o" removes the corresponding component from the overall framework.

	ICEWS14				ICEWS18			
model	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
REGCN	12.66	8.23	13.68	21.34	12.07	7.49	13.34	20.66
xERTE	16.9	13.8	19.4	22.67	8.05	6.33	9.29	11.39
T-logic	16.46	13.51	17.98	21.81	8.6	7.2	9.00	11.08
PPT	10.87	7.93	11.30	16.35	3.22	1.90	3.49	5.34
HGLS	19.92	13.69	22.16	33.35	14.44	7.7	15.07	29.26
w/o LLM	29.63	22.67	31.78	43.44	25.85	19.89	26.80	37.58
w/o FLM	16.75	10.38	18.56	32.42	8.44	3.95	8.23	18.12
w/o FEM	11.68	6.96	12.76	20. 64	10.88	6.46	12.03	19.04
w/o LEM	30.94	23.20	33.58	46.22	26.70	20.81	28.13	37.96
FL-Evo	**31.04**	**23.20**	**33.81**	**47.56**	**26.94**	**21.07**	**28.18**	**38.03**
	+11.12	+9.40	+11.65	+14.21	+14.87	+13.58	+14.84	+17.37

Table 5
The seen entity prediction results on ICEWS14 and ICEWS18. All results are percentages. We rerun their codes with the released hyperparameters. The best results are in bold. "w/o" removes the corresponding component from the overall framework.

	ICEWS14				ICEWS18			
Model	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
REGCN	43.74	33.01	49.06	64.30	33.25	22.85	37.43	53.73
xERTE	42.6	34.12	47.90	59.86	30.23	21.58	34.64	48.13
T-logic	44.92	34.99	50.44	63.94	30.76	21.18	34.93	50.00
PPT	39.84	29.73	44.39	59.67	26.67	16.69	30.52	46.58
HGLS	47.76	35.31	53.95	72.66	29.47	18.71	32.80	51.55
w/o LLM	49.10	36.38	55.78	73.72	34.54	23.49	38.84	56.87
w/o FLM	47.31	34.76	53.75	72.09	33.20	22.70	36.70	53.75
w/o FEM	39.22	28.53	44.30	59.74	33.12	23.59	36.74	51.61
w/o LEM	48.73	36.16	55.23	73.12	35.17	24.20	39.19	56.82
FL-Evo	**50.04**	**37.04**	**57.04**	**75.26**	**35.81**	**24.67**	**40.61**	**57.52**
	+2.28	+1.73	+3.09	+2.60	+2.56	+1.76	+3.18	+3.79

Table 6
Ablation Study on ICEWS14 and ICEWS18. All results are percentages. "w/o" removes the corresponding component from the overall framework.

	ICEWS14				ICEWS18			
Model	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
w/o LLM	46.78	34.57	53.09	70.31	34.36	23.32	38.53	56.54
w/o FLM	44.35	32.51	50.26	67.72	32.43	22.24	35.85	52.64
w/o FEM	37.43	27.13	42.24	57.20	32.89	23.46	36.51	51.44
w/o LEM	46.47	34.43	52.57	69.81	34.85	24.02	38.83	56.46
FL-Evo	48.09	35.63	55.00	71.90	35.43	24.47	39.62	57.10

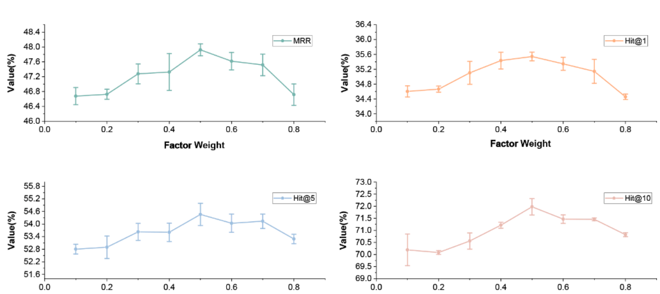


Fig. 7. The sensitivity analysis of the factor weight α on ICEWS14.

e.g. red signifies the entity belonging to Greece, including *Kidnapper(Greece)*, *Citizen(Greece)*, *Lawmaker(Greece)*, etc. Comparing Fig. 8(a) and (b), embeddings initialized by LLM outperform random initialization. Entities sharing similar concepts are clustered closer together, like the *Head of Government* of *Greece*, *Pakistan*, and *Botswana*. The entities with *China* are surrounding the entity *China*. It demonstrates LLM provides the concept of entities. As illustrated in Fig. 8(c), FLM encoding is the best, i.e., each group in its zone, refining the embedding generated by the LLM. It denotes the comprehensive concept of entities should be across the entire time, which can improve the reasoning performance.

In addition, we analyze the entity embeddings for entities without history, represented in brown. From Fig. 8(b), the entities with *Greece* are close to the Greece group, demonstrating LLM contains knowledge about Greece assisting in entity embedding. As shown in Fig. 8(c), all unseen entities are clustered within the *Greece* group, highlighting the effectiveness of FLM in encoding unseen entities. Therefore, learning entity representations across the historical context is crucial for comprehensive knowledge representation and improved reasoning performance.

4.6. *Ablation study*

To evaluate the impact of each component on FL-Evo, we conduct ablation experiments on the ICEWS14 and ICEWS18 datasets. As depicted in Table 6, each component has a positive effect on our model. FEM exhibits the greatest influence on the results, indicating the significance of capturing the fact evolution based on entity-based subgraph sampling

(a) Entity embedding after random initialization (b) Entity embedding after LLM encoding (c) Entity embedding after FLM encoding

Fig. 8. The visualization of embedding entities on ICEWS14. Each color represents the corresponding group. For example, the red indicates the entities belonging to Greece, including Policy(Greece), Citizen(Greece), Lawmaker(Greece), etc. The brown denotes the unseen entity embeddings.

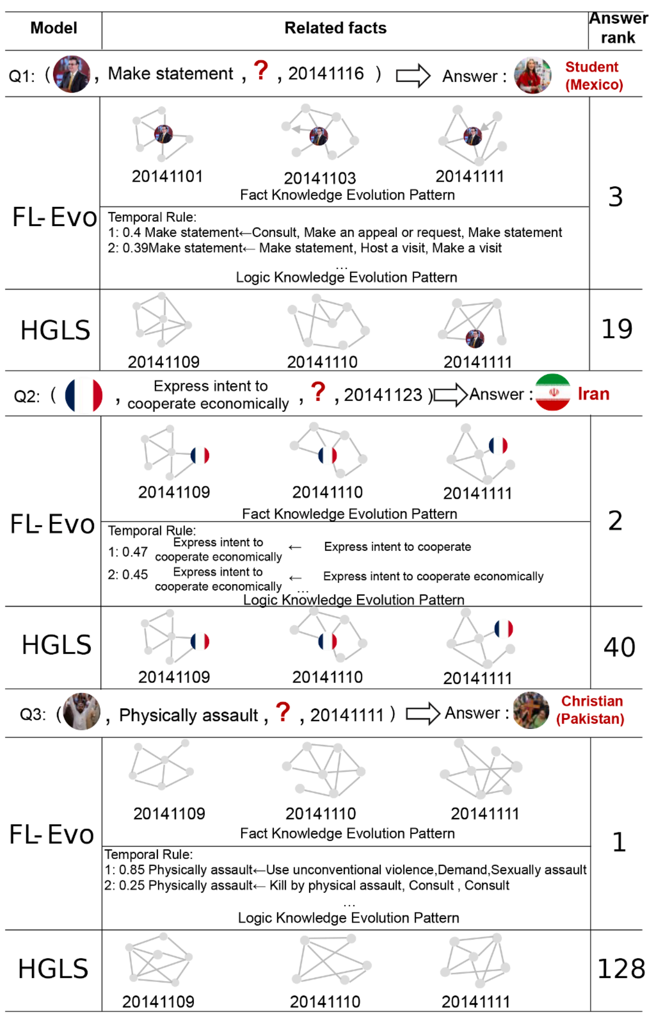


Fig. 9. The case study of HGLS and FL-Evo. The red font denotes the answer. The picture represents the source entity. Each temporal subgraph is annotated with a timestamp. In LEM, each number means the confidence of the rule. $r_q \leftarrow r_1, \ldots, r_n$ is equal to $r_q(e_s, e_o, t_q) \leftarrow r_1(e_s, z_1, t_1), \ldots, r_n(z_{n-1}, e_o, t_n)$, where $t_1, \ldots, t_n < t_q$. Q3 is the reasoning process of the unseen entity.

strategy. Conversely, continuous sampling fails to obtain the fact evolution. The FLM consistently contributes to the reasoning performance across datasets, demonstrating the benefits of utilizing existing information for extrapolation reasoning. Even though the LLM contains the bias, the integration of world knowledge from LLM enhances the representation of entities and relations, boosting reasoning performance. Additionally, LEM also plays an important role in FL-Evo, illustrating that logic knowledge evolution can effectively assist in performing predictions.

4.7. Case study

To illustrate the effectiveness of the proposed model, Fig. 9 presents the reasoning process of HGLS and FL-Evo. Comparing the results, four points can be concluded: (1) Improving the reasoning performance. FL-Evo consistently ranks ground truth higher than HGLS, indicating enhanced predictive accuracy. (2) Enhancing evolution modeling. As shown in Fig. 9 Q1, the related facts are not constantly occurring, i.e., *20141101, 20141103 and 20141111* [6]. FEM effectively samples the related subgraphs and captures the fact evolution pattern. (3) Leveraging knowledge sufficiently. In Q2, given a query *(France, Express intent to cooperate economically, ?, 20141123)*, despite HGLS sampling subgraphs containing relevant facts, the result is still lower than the proposed model. FL-Evo leverages information from various aspects to capture evolution patterns, improving reasoning performance. (4) Enriching the entity representation. Q3 is the reasoning performance of an unseen entity. The proposed model represents the events across the entire timeline and draws the world knowledge from LLM. As shown in Fig. 8 (c), the proposed model represents the unseen entity accurately, which is useful for prediction.

In summary, FLM contains sufficient knowledge that can be used in FEM and LEM. FEM samples relevant subgraphs to learn fact evolution and perform predictions. LEM realizes the prediction via grounding the temporal rules. FM achieves the final result through fusing these two evolution patterns. These cases demonstrate the ability of each component and the effectiveness of the proposed model.

5. Conclusions and future work

In this paper, FL-Evo improves reasoning performance by capturing fact and logic knowledge evolution patterns. The proposed model includes four parts, including fact and logic knowledge (FLM), fact evolution (FEM), logic evolution (LEM), and fusion module (FM). The FLM serves as background knowledge base, incorporating fact and logic knowledge derived from LLMs and TKG. For fact knowledge evolution pattern, the fact knowledge formed by entity and relation is first distilled from LLMs using designed prompts, followed by refined with TKG. FEM then extracts entity-based subgraphs from FLM, aiding in capturing fact evolution patterns. Additionally, LEM utilizes the logical knowledge,

[6] In this paper, YYMMDD is equal to YY.MM.DD

mined from FLM, to derive the logical knowledge evolution pattern, assisting in reasoning. Finally, the fusion module enhances the reasoning performance by integrating the outputs of these two evolution patterns. Extensive experiments have been conducted on five benchmarks. The experiment results outperform the TKG reasoning models, with improvements of up to 3.97 % and 4.07 % in Hit@3 and Hit@10, respectively. It is particularly noteworthy that FL-Evo has substantially enhanced reasoning performance for unseen entities lacking prior records, exceeding baseline models by 14.87 % in MRR and 17.37 % in Hit@10.

Moreover, our model also faces several challenges. First, we leverage LLMs to enhance entity and relation embeddings, and LLMs contain biases that can impact reasoning performance. Future research should develop effective methods to identify and mitigate biases, improving the accuracy of the embeddings. Second, the proposed sampling strategy relies on a fixed length and exponential distribution, which limits its ability to extract relevant facts comprehensively. To address this limitation, future work could explore alternative approaches, such as reinforcement learning and causal inference, to enable adaptive sampling. By treating the sampling process as a sequential decision problem, reinforcement learning could be employed. The state, action space, reward function, and policy network of reinforcement learning could be developed to guide the model in capturing relevant facts comprehensively, improving reasoning performance. Third, the framework of our proposed model is complex, and there is a need for a more intuitive tool to simplify its application. Lastly, it is crucial to test the model across different domains to evaluate its generalizability and practical utility.

Data availability

I have shared the link to my code in the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ruishen Liu: Conceptualization, Methodology, Formal analysis, Data curation, Visualization, Writing – original draft, Writing – review & editing, Resources, Investigation; **Xinzhi Wang:** Formal analysis, Conceptualization, Visualization, Writing – original draft, Writing – review & editing, Funding acquisition, Supervision; **Shaorong Xie:** Methodology, Writing – original draft, Writing – review & editing, Funding acquisition, Supervision; **Xiangfeng Luo:** Funding acquisition, Writing – review & editing, Supervision; **Huizhe Su:** Methodology, Writing – original draft, Writing – review & editing.

Acknowledgments

This work was supported in part by the Outstanding Academic Leader Project of Shanghai under the grant No.20XD1401700, the National Natural Science Foundation of China under the grant No. 91746203, and the National Natural Science Foundation of China under the grant No. 72204155.

Appendix A. Prompt example

In the proposed model, we design various prompts to extract world knowledge from a large language model, enriching the representation of entities and relations. These prompts are customized for entities and relations to maintain data integrity and confidentiality. For example, if the query timestamp is 2014, the entity prompt is *'Before 2014, what's the background knowledge of the entity [entity]?'*. For the relation, the prompt is *'what's the meaning of the relation [relation]'*.

Table A.1
The example of prompt for relation information extraction on ICEWS14. $relation^{-1}$ means the inverse relation.

Relation	Prompt Template	LLM response
Assassinate	Template: what is the meaning of the relation '[relation]'? Example: what is the meaning of the relation 'Assassinate'?	Assassinate is a verb that means to kill someone secretly or by treachery, especially a political leader or a public figure. It can also refer to the act of attempting to kill someone in this way.
$Assassinate^{-1}$	Template: what is the inverse meaning of the relation '[relation]'? Example: what is the inverse meaning of the relation 'Assassinate'?	The opposite or inverse meaning of the word "assassinate" is "protect" or "defend". … The police department worked to defend the community from potential threats and attacks.

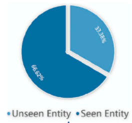
Table A.2
The example of prompt for entity information extraction on ICEWS14 and ICEWS18.

DataSet	Prompt Template	LLM response
ICEWS14	Template: before 2014, what is the background knowledge of the entity [Entity] Example: before 2014, what is the background knowledge of the entity Joaquim Barbosa?	Joaquim Barbosa is a Brazilian lawyer and politician who served as a Federal SupFEMe Court (STF) justice from 2003 to 2018. He was born in 1950 in the state of Bahia, … He has been a vocal critic of corruption and inequality in Brazil.
ICEWS18	Template: before 2018, what is the background knowledge of the entity [Entity]? Example: before 2018, what is the background knowledge of the entity Alassane Ouattara?	Alassane Ouattara is a politician and economist from Côte d'Ivoire who has been the President of Côte d'Ivoire since 2010 … After the 2010 presidential election, in which Ouattara defeated Gbagbo, he was sworn in as President of Côte d'Ivoire.

Tables A.1 and A.2 list the relation and entity prompts used in the proposed model. In our experiments, we refer to LLAMAv2-13B as LLM and utilize its encoder to encode responses. These embeddings serve as the initial representation for both entities and relations. For example, if the entity is *'Alassane Ouattara'* and the timestamp is 2018. The prompt is *'before 2018, what is the background knowledge of the entity Alassane Ouattara?'*. The LLM generates the response based on the prompt, like *'Alassane Ouattara is a politician and economist from Côte d'Ivoire who … President of Côte d'Ivoire'*. After LLM answers, the response is fed into LLM again to get the corresponding encoding based on the hidden state. The encoding is used to initialize the reasoning process. The process of relation is similar to the entity. In addition, we randomly extract 100 responses generated by the LLM for manual evaluation to ensure the reliability of the generated responses. To speed up the reasoning process, the LLM responses can be stored in advance.

Appendix B. Test dataset

In this paper, we conduct extrapolation temporal reasoning. Due to the never-ending information and the limitation of information extraction methods, not every entity has a record. As shown in Fig. B.1, in ICEWS14 and ICEWS18, there are 25.79 % and 33.38 % entities without history. The key to improving reasoning performance is to capture the evolution patterns. If an entity does not have a record, it is difficult to capture the entity evolution pattern, leading to weakened reasoning performance. In this paper, we leverage the known information from diverse aspects to model knowledge evolution. And we conduct



Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	Train	Valid	Test	Unseen	Seen
ICEWS14	7,128	230	63,685	13,823	13,222	862	12,360
ICEWS18	23,033	256	373,018	45,995	49,545	1,948	4,7597

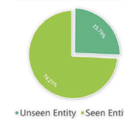


Fig. B.1. The details of unseen and seen dataset. $|\mathcal{E}|$ and $|\mathcal{R}|$ are the number of entities and relations respectively. Unseen and seen denote the size of unseen and seen test datasets respectively. The Pie charts denote the unseen entity and seen entity ratio in each dataset.

experiments on unseen entities to prove the effectiveness of our model. Fig. B.1 presents the dataset used to conduct unseen and seen entity experiments (Section 4.3). We only split the original test dataset into two parts, i.e., unseen and seen queries. The original train dataset is used to train the proposed model. According to the released codes and the hyperparameters, we rerun the baselines to get their results.

Appendix C. Examples

To demonstrate the reasoning process of our model and its application across different domains, we present additional examples (Fig. C.1). The first example is from the financial dataset (FinDKG), while the second is political event prediction contained in ICEWS14. In FEM, the entity-based sampling strategy is employed to extract subgraphs containing the source entity, i.e., 2022.06.12, 2022.06.19, and 2022.06.26 in Q1, and 2014.09.16, 2014.09.17, and 2014.09.18 in Q2. Using the extracted subgraphs and representations derived from FLM, we make predictions based on fact evolution patterns. Then, we use the rules automatically extracted from TKG through rule mining methods to realize the logic evolution prediction. Finally, the fusion module integrates outputs from these evolution patterns to predict future events.

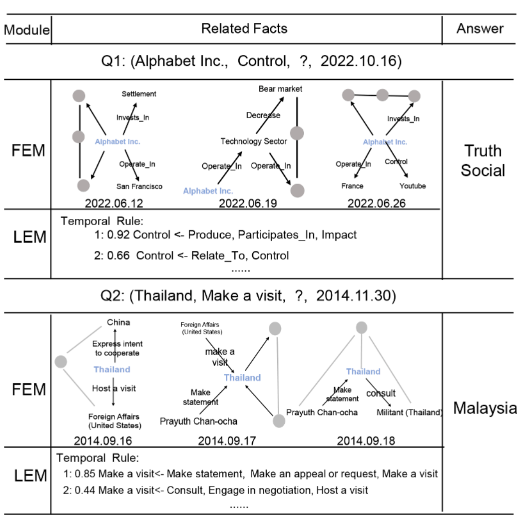


Fig. C.1. The case study on the financial and politics event prediction . The blue denotes the source entity.

References

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in Neural information processing Systems*, *26*, 2787–2795.

Boschee, E., Lautenschlager, J., O'Brien, S., Shellman, S., Starz, J., & Ward, M., (2015). ICEWS Coded Event Data. URL https://doi.org/10.7910/DVN/28075

Chen, T., Long, J., Wang, Z., Luo, S., Huang, J., & Yang, L. (2024). THCN: A Hawkes process based temporal causal convolutional network for extrapolation reasoning in temporal knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, *36*(12), 9374–9387. https://doi.org/10.1109/TKDE.2024.3474051

Deng, S., Rangwala, H., & Ning, Y. (2020). Dynamic knowledge graph based multi-event forecasting. *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. https://api.semanticscholar.org/CorpusID:221191715.

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2D knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*. *(vol. 32)*.

Gao, Y., Feng, L., Kan, Z., Han, Y., Qiao, L., & Li, D. (2022). Modeling precursors for temporal knowledge graph reasoning via auto-encoder structure. In *International joint conference on artificial intelligence*. https://api.semanticscholar.org/CorpusID:250630286.

García-Durán, A., Dumancic, S., & Niepert, M. (2018). Learning sequence encoders for temporal knowledge graph completion. In *Conference on empirical methods in natural language processing*. https://api.semanticscholar.org/CorpusID:52183483.

Goel, R., Kazemi, S. M., Brubaker, M. A., & Poupart, P. (2020). Diachronic embedding for temporal knowledge graph completion. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020* (pp. 3988–3995). AAAI Press. https://doi.org/10.1609/aaai.v34i04.5815. https://doi.org/10.1609/AAAI.V34I04.5815

Han, Z., Chen, P., Ma, Y., & Tresp, V. (2021). Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International conference on learning representations*. https://api.semanticscholar.org/CorpusID:235614395.

Jiang, T., Liu, T., Ge, T., Sha, L., Chang, B., Li, S., & Sui, Z. (2016). Towards time-aware knowledge graph completion. In *International conference on computational linguistics*. https://api.semanticscholar.org/CorpusID:13475624.

Jin, W., Qu, M., Jin, X., & Ren, X. (2019). Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *Conference on empirical methods in natural language processing*. https://api.semanticscholar.org/CorpusID:222205878.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lacroix, T., Obozinski, G., & Usunier, N. (2020). Tensor decompositions for temporal knowledge base completion. In *8th International conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net. https://openreview.net/forum?id=rke2P1BFwS.

Li, N., E, H., Li, S., Sun, M., Yao, T., Song, M., Wang, Y., & Luo, H. (2023). TR-rules: Rule-based model for link forecasting on temporal knowledge graph considering temporal redundancy. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Findings of the association for computational linguistics: EMNLP 2023* (pp. 7885–7894). Singapore: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.529

Li, X. V., & Sanna Passino, F. (2024). FinDKG: Dynamic knowledge graphs with large language models for detecting global trends in financial markets. In *Proceedings of the 5th ACM international conference on AI in finance* ICAIF '24 (p. 573-581). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3677052.3698603

Li, Y., Sun, S., & Zhao, J. (2022a). TiRGN: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *International joint conference on artificial intelligence*. https://api.semanticscholar.org/CorpusID:250631397.

Li, Z., Hou, Z., Guan, S., Jin, X., Peng, W. B., Bai, L., Lyu, Y., Li, W., Guo, J., & Cheng, X. (2022b). HiSMatch: Historical structure matching based temporal knowledge graph reasoning. *ArXiv*, *abs/2210.09708*. https://api.semanticscholar.org/CorpusID:252967878.

Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., & Cheng, X. (2021). Temporal knowledge graph reasoning based on evolutional representation learning. *Proceedings of the 44th International ACM SIGIR conference on research and development in information retrieval*. https://api.semanticscholar.org/CorpusID:233324265.

Lin, Q., Liu, J., Mao, R., Xu, F., & Cambria, E. (2023). TECHS: Temporal logical graph networks for explainable extrapolation reasoning. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1281–1293). Toronto, Canada: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.71

Liu, X., Zhang, J., Ma, C., Liang, W., Xu, B., & Zong, L. (2024). Temporal knowledge graph reasoning with dynamic hypergraph embedding. In *International conference on language resources and evaluation*. https://api.semanticscholar.org/CorpusID:269803983.

Liu, Y., Ma, Y., Hildebrandt, M., Joblin, M., & Tresp, V. (2022). TLogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(4), 4120–4127. https://doi.org/10.1609/aaai.v36i4.20330

Ma, J., Li, K., Zhang, F., Wang, Y., Luo, X., Li, C., & Qiao, Y. (2024). TaReT: Temporal knowledge graph reasoning based on topology-aware dynamic relation graph and temporal fusion. *Information Processing & Management*. https://api.semanticscholar.org/CorpusID:271525801.

Meilicke, C., Chekol, M. W., Ruffinelli, D., & Stuckenschmidt, H. (2019). Anytime bottom-up rule learning for knowledge graph completion. In S. Kraus (Ed.), *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI 2019, macao,* *china, august 10-16, 2019* (pp. 3137–3143). ijcai.org. https://doi.org/10.24963/IJCAI.2019/435

Omran, P. G., Wang, K., & Wang, Z. (2019). Learning temporal rules from knowledge graph streams. In *AAAI Spring symposium combining machine learning with knowledge engineering*. https://api.semanticscholar.org/CorpusID:135466806.

Schlichtkrull, M., Kipf, T., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2017). Modeling relational data with graph convolutional networks. In *Extended semantic web conference*. https://api.semanticscholar.org/CorpusID:5458500.

Sun, H., Geng, S., Zhong, J., Hu, H., & He, K. (2022). Graph hawkes transformer for extrapolated reasoning on temporal knowledge graphs. In *Conference on empirical methods in natural language processing*. https://api.semanticscholar.org/CorpusID:256461300.

Sun, H., Zhong, J., Ma, Y., Han, Z., & He, K. (2021). Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. *ArXiv, abs/2109.04101*. https://api.semanticscholar.org/CorpusID:237454564.

Tone, A. (2015). Global data on events, location and tone (GDELT). https://api.semanticscholar.org/CorpusID:218095614.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S. et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning* (pp. 2071–2080). PMLR.

Wan, G., Pan, S., Gong, C., Zhou, C., & Haffari, G. (2020). Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In *Twenty-ninth international joint conference on artificial intelligence and seventeenth pacific rim international conference on artificial intelligence IJCAI-PRICAI-20*.

Xiong, W., Hoang, T., & Wang, W. Y. (2017). DeepPath: A reinforcement learning method for knowledge graph reasoning. In M. Palmer, R. Hwa, & S. Riedel (Eds.), *Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, copenhagen, denmark, september 9-11, 2017* (pp. 564–573). Association for Computational Linguistics. https://doi.org/10.18653/v1/d17-1060

Xu, W., Liu, B., Peng, M., Jia, X., & Peng, M. (2023). Pre-trained language model with prompts for temporal knowledge graph completion. In *Annual meeting of the association for computational linguistics*. https://api.semanticscholar.org/CorpusID:258686648.

Xu, Y., Ou, J., Xu, H., & Fu, L. (2022). Temporal knowledge graph reasoning with historical contrastive learning. In *AAAI Conference on artificial intelligence*. https://api.semanticscholar.org/CorpusID:253734921.

Zhang, J., Hui, B., Mu, C., & Tian, L. (2024). Learning multi-graph structure for temporal knowledge graph reasoning. *Expert Systems with Applications*, *255*, 124561. https://doi.org/10.1016/j.eswa.2024.124561

Zhang, M., Xia, Y., Liu, Q., Wu, S., & Wang, L. (2023). Learning long- and short-term representations for temporal knowledge graph reasoning. *Proceedings of the ACM Web Conference 2023*. https://api.semanticscholar.org/CorpusID:258333948.

Zhao, M., Zhang, L., Kong, Y., & Yin, B. (2021). Temporal knowledge graph reasoning triggered by memories. *ArXiv, abs/2110.08765*. https://api.semanticscholar.org/CorpusID:239016656.

Zhen, M., Wang, J., Zhou, L., Fang, T., & Quan, L. (2018). End-to-end structure-aware convolutional networks for knowledge base completion. *Proceedings of the ... AAAI conference on artificial intelligence. AAAI conference on artificial intelligence*, *33*, 3060–3067. https://api.semanticscholar.org/CorpusID:262690390.

Zhu, C., Chen, M., Fan, C., Cheng, G., & Zhan, Y. (2020). Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *AAAI Conference on artificial intelligence*. https://api.semanticscholar.org/CorpusID:229180723.