2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00050

IE-Evo: Internal and External Evolution-Enhanced Temporal Knowledge Graph Forecasting

Kangzheng Liu[†‡], Feng Zhao[†*], Guandong Xu[‡], and Shiqing Wu[‡]
[†]Natural Language Processing and Knowledge Graph Lab,
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[‡]Data Science and Machine Intelligence Lab, University of Technology Sydney, Sydney, Australia
[†]{frankluis, zhaof}@hust.edu.cn, [‡]{guandong.xu, shiqing.wu}@uts.edu.au

***Abstract*—Temporal knowledge graph (TKG) forecasting is widely used in various fields due to its ability to infer future events based on historical information. Modeling the internal structures and chronological dependencies of historical subgraph sequences has been proven effective. Nevertheless, on the one hand, the TKG forecasting process generally suffers from a lack of sufficient sample data due to historical resource limitations; thus, most works focus on continuously mining the patterns of historical sequences while ignoring the semantically-rich background information provided by external knowledge, especially when historical query-related information is scarce. On the other hand, when merely serializing the given subgraph sequence to mimic its temporal evolution process, only the chronological dependencies between the subgraphs can be considered, thus ignoring the evolution of time information. Hence, a method that integrates internal and external knowledge to enhance the representations of entities is urgently needed. To this end, we propose a novel TKG forecasting method, namely, the internal and external evolution-enhanced framework (IE-Evo). For the former issue, we design an external evolution encoder and use a pre-trained language model (PLM) to provide powerful external knowledge semantics for TKG forecasting. To address the latter concern, we propose an internal evolution encoder that explicitly embeds the time information while modeling the aggregation and evolution processes of the observed sequential structural information. IE-Evo has been evaluated on four public benchmark datasets, showcasing its significant improvements across multiple evaluation metrics.**

***Index Terms*—Temporal knowledge graph extrapolation, External knowledge, Time information evolution**

I. Introduction

Temporal knowledge graphs (TKGs) incorporate time information into traditional knowledge graphs and represent a real-world fact (event) as a quadruple ($subject$, $relation$, $object$, $timestamp$). A TKG is actually composed of static subgraphs divided by the time dimension, and each subgraph contains all the facts occurring at the specific corresponding timestamp. TKG forecasting focuses on inferring incomplete events for a future subgraph according to the information contained in the historical subgraphs. Given its practical significance, TKG forecasting finds extensive applications in domains such as financial forewarning and behavioral prediction. To obtain more information for forecasting future events, an increasing number of works [1]–[5] have focused on mining the patterns of historical subgraphs inside TKGs. This is actually a graph sequence modeling problem, where not only the structural information within each subgraph but also the temporal evolution of the subgraph sequence need to be taken into account. However, two main challenges remain to be addressed.

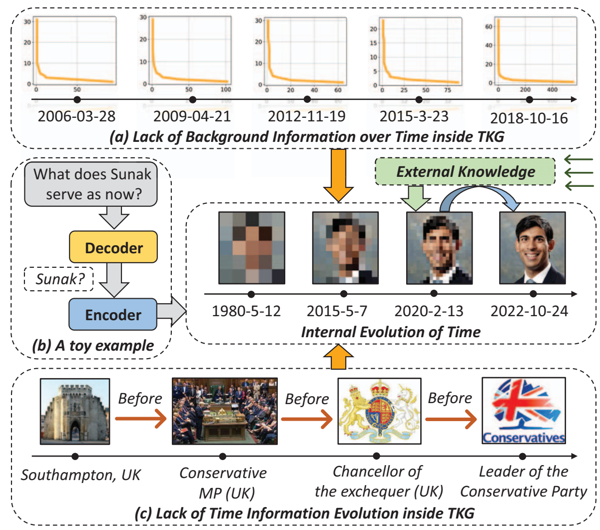

Fig. 1. Illustration of the issues regarding the lack of background information and time information evolution inside a TKG for obtaining clear entity portraits of interest.

The lack of background information over time inside a TKG. The historical space can be regarded as a background knowledge base in nature and the challenge regarding the lack of background information is always a limitation of the modeled historical scope. As shown in Figure 1(a), in the event-based TKG Integrated Crisis Early Warning System, we analyze the dynamic relationship between the frequencies of entities and their numbers of associated facts. It is commonly observed that the majority of entities have sparse associated quadruples over time. Consequently, in the long term, a TKG lacks sufficient historical information to effectively forecast future events. Thus, persistently mining the internal evolution process may yield little information; nevertheless, few works thus far have focused on the external knowledge outside a TKG. In the well-adopted codec-based architecture, the encoder aims to make the embedding of an entity contain

*Corresponding author

979-8-3503-0788-7/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDM58522.2023.00050

as much useful entity-related information as possible in the structural aggregation and evolution processes to obtain an accurate representation. This process can be similarly described as drawing an entity portrait.

Then, Figure 1(b) illustrates the decoding process based on the different degrees of entity portraits. When answering (Sunak, serves as, ?, 2023-9-1), the decoder first asks the encoder "who is Sunak?", which means obtaining the representation of the "Sunak" entity. We define the background information as an event text that is relevant to the entity of interest. On the one hand, if "Sunak" is a newly emerging entity, as the first portrait shows, the lack of background information makes the encoder completely unable to answer the question, and the vague description makes the decoder produce a forecast with an almost random guess. Thus, external knowledge can undoubtedly help the encoder learn the complete entity representation. On the other hand, if "Sunak" has rich repetitive patterns in history, e.g., as the third portrait shows, numerous facts, such as (Sunak, was born in, Southampton (UK), 1980-5-12) and (Sunak, served as, Conservative Member of Parliament (MP), 2015-5-7), appeared in history, then the encoder's representation of the "Sunak" entity becomes clearer, but the additional background information provided by external knowledge can also enhance the entity embedding. A clear portrait of the entity in the encoder is helpful for obtaining the query's solution in the decoder.

The lack of time information evolution inside a TKG. The temporal evolution of a subgraph sequence includes the evolution of time information in addition to the chronological dependencies between different pieces of structural information. Sequential modeling only considers the chronological dependencies of a TKG; that is, the model can only understand which timestamp occurs former and which occurs latter, but cannot know the exact time of an event, which may result in an inaccurate judgment of a temporal evolution process containing long-term patterns. As shown Figure 1(c), in a case where only sequential modeling is performed, the model does not take the specific representations of time information such as 1980-5-12, 2015-5-7, 2020-2-13, and 2022-10-24 into account. Instead, it only knows that Sunak served as the British chancellor of the exchequer before serving as the leader of the British Conservative Party, as a Conservative MP before becoming the chancellor of the exchequer, and was born in Southampton before becoming an MP. If Sunak was born in the 19^{th} century, it would make no sense to forecast what he will serve as on September 1^{st}, 2023. Therefore, merely modeling the sequential information without considering the evolution of time information would greatly reduce the limited aggregatable background information that the historical space can provide. Thus, current works not only completely ignore the evolution of external knowledge, but also fail to completely model the temporal evolution of internal knowledge.

Our contributions. In this paper, we propose a TKG forecasting method, namely, the **I**nternal and **E**xternal **Evo**lution-enhanced framework (IE-Evo), to address the abovementioned challenges. As shown in Figure 2, for the former issue, we utilize a pre-trained language model (PLM) in the external evolution encoder to introduce external knowledge related to the entities. Specifically, we utilize each entity's textual references in the real world to establish a link with the external PLM and then introduce external background information to enhance the representations of entities. We can then obtain the initial embeddings, which aggregate the external background information of the corresponding entities. Then, we model the structural aggregation and the chronological dependencies of the external knowledge via a GCN and an RNN, respectively. For the latter issue, to effectively utilize the limited historical information within a TKG, we need to simultaneously consider the chronological dependencies and time information evolution, thus enhancing the temporal evolution modeling of the internal knowledge. As Figure 2 shows, we design an internal evolution encoder, in which a time cell is designed to address the problem of time information evolution, and an RNN is used to model the chronological dependencies. It explicitly embeds and updates the timestamp information of the next historical subgraph in the temporal development process, thus making the time information and structure information participate in the evolution of the historical sequence together.

The contributions of our work are summarized as follows.

- We propose an internal and external evolution-enhanced framework for TKG forecasting, which comprehensively integrates entity-related information within and outside the TKG to enhance the representations of entities for future event forecasting.
- In the framework, we design two encoders that perform external and internal evolution to address the shortage of background information and the lack of time information evolution, respectively, in the historical subgraph sequence of a TKG.
- Substantial experiments are carried out on four well-known TKG datasets. The effectiveness of IE-Evo for TKG forecasting is evident from the improvements observed across almost all performance metrics.

The remainder of this paper is organized as follows. Section II discusses the related work. Section III details the IE-Evo model. Experimental analyses are contained in Section IV, followed by the conclusions in Section V.

II. Related Work

There are mainly two modeling strategies currently utilized in TKG forecasting.

a) Static Forecasting Methods: Temporal dynamics are not considered in this modeling strategy. Translation-based methods leverage a low-dimensional vector space to embed both relations and entities; these approaches include RotatE [6], etc. Among them, RotatE implements relation-based rotation from subjects to objects. Matrix decomposition-based methods model a specific relation as a decomposable matrix; such approaches include ComplEx [7], DistMult [8], etc. Convolution-based methods define entities and relations as matrices to conduct the operations of convolution; the representative models include Conv-TransE [9] and ConvE [10].

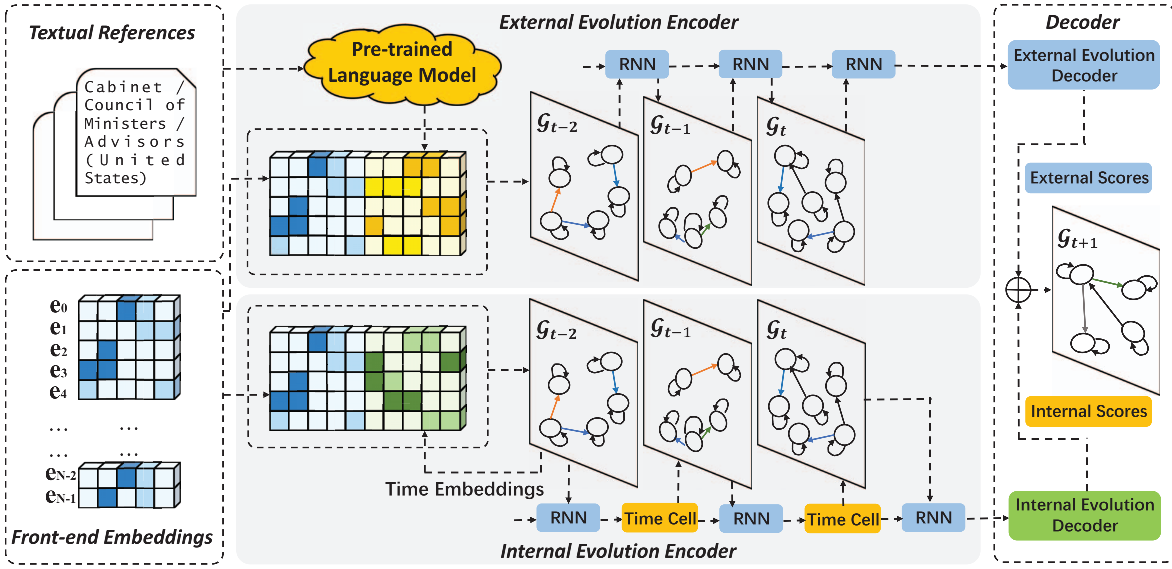

Fig. 2. The framework of IE-Evo. The blue matrices indicate the front-end part of the entity embeddings. The yellow and green matrices represent the back-end part of the entity embeddings derived from the pre-trained language model (PLM) and the time information, respectively.

GCN-based methods, including R-GCN [11], can aggregate adjacent information of the entities via the message-passing architecture for a static KG.

b) Dynamic Forecasting Methods: This kind of modeling strategy can be divided into two categories according to whether the utilized method is suitable for scenarios involving the inference of future events: interpolation and extrapolation. Interpolation methods are trained under the condition that the global information (including historical and future information) is known, and thus, these approaches, such as HyTE [12], TA-DistMult [13], and TTransE [14], are not good at future forecasting tasks. In the extrapolation setting, only historical information is available. CyGNet [15] introduces a copy-generation mechanism derived from abstractive summarization. RE-NET [1] treats the historical information as conditional probabilities. xERTE [16] builds an inference graph with query entities as the center. TLogic [17] performs a temporal random walk through rule learning. CEN [3] adopts an online training strategy to cope with the time-varying challenge. TITer [18] calculates the future entity embeddings of a specific query path via reinforcement learning. DA-Net [19] and DHU-NET [20] copes with the time-variability evolution of entity representations. RETIA [21] aggregates complete neighborhood messages by constructing positional hyperrelation subgraphs. Following RE-GCN [2], TiRGN [4] also uses a recurrent relational GCN and an RNN to model the given historical sequence. HGLS [5] designs a hierarchical relational GCN to aggregate short-term and long-term structural information. CENET [22] employs a statistical-based method that obtains the representations of non-historical events and entities through contrastive learning.

Among the abovementioned extrapolation approaches, only xERTE, TITer, and CENET attempt to address the representation problem of entities with scarce background information. However, they are still stuck mining the hidden patterns inside a TKG and neglect to provide the missing semantics of internal information through external knowledge. PPT [23] converts the TKG forecasting task into a masked token prediction task through prompts, but it relies entirely on the external knowledge provided by PLMs, instead ignoring the structural aggregation and temporal evolution processes inside a TKG. On the other hand, some early interpolation methods and the latest extrapolation methods consider the time information representation problem. For example, HyTE regards a specific time embedding as a hyperplane. TiRGN incorporates time embeddings into the decoding process. HGLS models time embeddings in the constructed global KG. However, both interpolation and extrapolation approaches neglect the evolution of time information (embeddings) over time.

III. Methodology

In this section, we provide a comprehensive explanation of our IE-Evo model. We start by introducing the notations and definitions. Subsequently, we delve into the overall framework and various components of the model. Additionally, we cover the training strategy and analyze the time complexity.

A. Notations and Definitions

A TKG $\mathcal{G}$ is formally represented as a static subgraph sequence $\mathcal{G}$=$\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \cdots, \mathcal{G}_T\}$, where T is the size of the timestamp set $\mathcal{T}$. Then, we define the entity set as $\mathcal{E}$, the entity reference set as $\mathcal{E}_r$ (both have sizes of N), and the relation set as $\mathcal{R}$ with a size of R. A static subgraph $\mathcal{G}_t$ consists of all the fact triples (s, r, o) occurring at a specific timestamp t, where $\{s, o\} \in \mathcal{E}$, $r \in \mathcal{R}$, and $t \in \mathcal{T}$. We define the relation embeddings of a TKG $\mathcal{G}$ for the internal and external evolution encoders as $\mathbf{R}_{In}$ and $\mathbf{R}_{Ex}$, respectively, the two encoders' entity embeddings for a certain subgraph $\mathcal{G}_t$ as $\mathbf{E}_{In}^{\mathrm{t}}$ and $\mathbf{E}_{Ex}^{\mathrm{t}}$, and the embedding of a specific timestamp t as $\mathbf{T}^{\mathrm{t}}$. As Figure 2 shows, the dimensions of the front-end embeddings, external back-end embeddings, and internal back-end embeddings are defined as d_{fr}, d_{lm}, and d_{te}, respectively. In general, TKG forecasting aims to infer a missing object (s, r, ?, t+1) or a missing subject (?, r, o, t+1) occurring at a future subgraph $\mathcal{G}_{t+1}$ when the l-length historical subgraphs $\{\mathcal{G}_\tau | \ t\text{-}l\text{+}1 \leq \tau < t\text{+}1\}$ are known.

B. Framework Overview

The proposed IE-Evo model is composed of two encoders and a decoder. As the yellow matrix in Figure 2 shows, the external evolution encoder is responsible for generating the background knowledge embeddings via an external PLM according to the entities' textual references, splicing them with the randomly initialized front-end entity embeddings (as the blue matrix in Figure 2 shows) and then inputting them into a sequence of GCNs and RNNs to perform the external knowledge evolution process. As the green matrix in Figure 2 shows, the internal evolution encoder is responsible for splicing the timestamp embedding of each historical subgraph with the front-end entity embeddings, and then inputting them into a sequence of GCNs, RNNs, and time cells to perform the internal knowledge evolution procedure. Note that the same front-end entity embeddings are used to establish the association between the two encoders. We do not define an explicit source (internal or external) for the randomly initialized front-end embeddings, and the main distinction lies in the different roles played by the back-end embeddings. The decoder simultaneously learns the roles of the internal and external evolution processes in TKG forecasting in the form of score summing.

C. External Evolution Encoder

This module aims to obtain the external background knowledge and model its evolution process. In this paper, we choose the pre-trained BERT [24] model (specifically, the bert-base-uncased model[1]) as the external PLM, and the R-GCN [11] and GRU [25] models as the units of the GCN and RNN sequences, respectively. The real-world reference sentences of the entities are utilized to establish links between the external PLMs and the TKGs. For example, in the ICEWS14 dataset, the real-world reference of the 24^{th} entity is "North Atlantic Treaty Organization". Then the input format of the BERT model is "[CLS] North Atlantic Treaty Organization [SEP]" for the 24^{th} entity. Note that these real-world textual references of the entities are provided in the TKG datasets. The traditional approaches are to number and randomly initialize the textual reference information of entities to learn the embeddings of internal knowledge. However, we utilize this information to establish a link with the external PLMs and then introduce external background information to enhance the representation of entities.

[1]https://huggingface.co/bert-base-uncased

a) External Knowledge Acquisition: The entity reference set $\mathcal{E}_r$ records the corresponding objective existence of each entity in the real world. We obtain the back-end entity embeddings of the fused external knowledge by inputting the set as a whole into the BERT model:

$$\mathbf{E}_{\mathrm{LM}} = \mathbf{W}_{\mathrm{LM}}\mathrm{BERT}(\mathcal{E}_{\mathrm{r}}) + \mathbf{b}_{\mathrm{LM}} \tag{1}$$

where $\mathrm{BERT}(\mathcal{E}_{\mathrm{r}}) \in \mathbb{R}^{d_{BERT}\times N}$, $\mathbf{W}_{\mathrm{LM}} \in \mathbb{R}^{d_{lm}\times d_{BERT}}$, $\mathbf{b}_{\mathrm{LM}} \in \mathbb{R}^{1\times N}$, and $\mathbf{E}_{\mathrm{LM}} \in \mathbb{R}^{d_{lm}\times N}$. d_{BERT} is defined as the output dimensionality of the BERT model. Note that each entity reference sentence in the set $\mathcal{E}_r$ may contain a different number of tokens. Thus, we adopt padding mask technology [26] to address this problem.

b) External Structural Aggregation: For an l-length historical sequence, we should aggregate the adjacent structural information of the entities in each subgraph. In particular, we concatenate the back-end embeddings of external knowledge with the randomly initialized front-end embeddings and input them into the first historical subgraph $\mathcal{G}_{t-l+1}$ for aggregation:

$$\mathbf{E}_{Ex}^{\mathrm{input}} = Con(\mathbf{E}_{init}, \mathbf{E}_{\mathrm{LM}}) \tag{2}$$

where $\mathbf{E}_{init} \in \mathbb{R}^{d_{fr}\times N}$ and $\mathbf{E}_{Ex}^{\mathrm{input}} \in \mathbb{R}^{(d_{fr}+d_{lm})\times N}$. $\mathbf{E}_{init}$ denotes the front-end embeddings of entities, and *Con* represents the concatenation operation. For a specific historical subgraph $\mathcal{G}_t$, the aggregation process can be formally represented as:

$$\mathbf{E}_{Egcn}^{\mathrm{t}} = R\_GCN_{\mathrm{Ex}}(\mathbf{E}_{Ex}^{\mathrm{t}-1}, \mathbf{R}_{Ex}) \tag{3}$$

where $\mathbf{E}_{Egcn}^{\mathrm{t}}$, $\mathbf{E}_{Ex}^{\mathrm{t}-1} \in \mathbb{R}^{(d_{fr}+d_{lm})\times N}$, and $\mathbf{R}_{Ex} \in \mathbb{R}^{(d_{fr}+d_{lm})\times R}$. $\mathbf{E}_{Egcn}^{\mathrm{t}}$ is the output of an external R-GCN model at the t^{th} timestamp. $\mathbf{E}_{Ex}^{\mathrm{t}-1}$ is actually the output of the $(t\text{-}1)^{th}$ external GRU model and the finally collected external entity embeddings at the $(t\text{-}1)^{th}$ timestamp. Note that $\mathbf{E}_{Ex}^{\mathrm{t}-l} = \mathbf{E}_{Ex}^{\mathrm{input}}$ when calculating the GCN output $\mathbf{E}_{Egcn}^{\mathrm{t}-l+1}$ of the first $(t\text{-}l\text{+}1)^{th}$ historical subgraph. Specifically, the message-passing architecture is adopted to perform the aggregation operation at each historical timestamp:

$$\mathbf{e}_o^{l+1} = f\left(\sum_{r\in\mathcal{R}}\sum_{s\in\mathcal{E}_o^r}\frac{1}{c_{o,r}}\mathrm{W}_r^l(\mathbf{e}_s^l + \mathbf{r}^l) + \mathrm{W}_0^l\mathbf{e}_o^l\right) \tag{4}$$

where $\mathbf{e}_o^l$ and $\mathbf{e}_s^l$ represent the l^{th}-layer embeddings of entities o and s in each R-GCN model, respectively. The set $\mathcal{E}_o^r$ records all of the entities that are linked to the updated entity o via a specific relation r. $c_{o,r}$ is the size of $\mathcal{E}_o^r$, and $f(\cdot)$ denotes the reflected rectified linear unit (RReLU) function. W_r^l and W_0^l are learnable parameters for a normal or self-looping edge r, respectively.

c) External Chronological Dependencies Modeling: To model the external structure dependencies between subgraphs, we utilize a sequence of RNNs, as shown in Figure 2. The inputs of the t^{th} RNN (specifically, a GRU) are the $(t\text{-}1)^{th}$ RNN output and the t^{th} GCN output:

$$\mathbf{E}_{Ex}^{\mathrm{t}} = GRU_{\mathrm{Ex}}(\mathbf{E}_{Ex}^{\mathrm{t}-1}, \mathbf{E}_{Egcn}^{\mathrm{t}}) \tag{5}$$

where $\mathbf{E}_{Ex}^{\mathrm{t}}$, $\mathbf{E}_{Ex}^{\mathrm{t}-1}$, and $\mathbf{E}_{Egcn}^{\mathrm{t}} \in \mathbb{R}^{(d_{fr}+d_{lm})\times N}$. $\mathbf{E}_{Ex}^{\mathrm{t}}$ denotes the final external entity embeddings at the t^{th} historical timestamp.

D. Internal Evolution Encoder

This encoder aims to completely model the structural and temporal evolution processes within a TKG. Note that the temporal evolution includes not only the chronological dependencies but also the evolution of time information.

a) Internal Time Information-Aware Aggregation: We form an evolutionary design by adding time information to each subgraph's structural information aggregation process. For the first subgraph $\mathcal{G}_{t-l+1}$ of the l-length historical sequence, we concatenate the same front-end embeddings as those of the external evolution encoder with the time embeddings of the current timestamp, and then use them as inputs for the internal GCN sequence:

$$\mathbf{E}_{In}^{\text{input}} = Con(\mathbf{E}_{init}, BC(\mathbf{T}^{\text{t}-l+1})) \tag{6}$$

where $\mathbf{E}_{In}^{\text{input}} \in \mathbb{R}^{(d_{fr}+d_{te})\times N}$ and $\mathbf{T}^{\text{t}-l+1} \in \mathbb{R}^{d_{te}}$. BC denotes the broadcasting operation that extends the shape of the vector $\mathbf{T}^{\text{t}-l+1}$ to $d_{te} \times N$. Based on the message-passing architecture described by Equation 4, the internal time information-aware aggregation process can be formally represented as:

$$\mathbf{E}_{Igcn}^{\text{t}-1} = R\_GCN_{\text{In}}(\mathbf{E}_{time}^{\text{t}-1}, \mathbf{R}_{In}) \tag{7}$$

where $\mathbf{E}_{Igcn}^{\text{t}-1}$, $\mathbf{E}_{time}^{\text{t}-1} \in \mathbb{R}^{(d_{fr}+d_{te})\times N}$ and $\mathbf{R}_{In} \in \mathbb{R}^{(d_{fr}+d_{te})\times R}$. $\mathbf{E}_{Igcn}^{\text{t}-1}$ and $\mathbf{E}_{time}^{\text{t}-1}$ are the outputs of an internal R-GCN model and a time cell at the $(t\text{-}1)^{th}$ timestamp, respectively. Note that $\mathbf{E}_{time}^{\text{t}-l} = \mathbf{E}_{In}^{\text{input}}$ at the first $(t\text{-}l\text{+}1)^{th}$ historical timestamp.

b) Internal Chronological Dependencies Modeling: A sequence of GRUs is also adopted to capture the chronological dependencies between the subgraphs within a TKG:

$$\mathbf{E}_{In}^{\text{t}-1} = GRU_{\text{In}}(\mathbf{E}_{time}^{\text{t}-1}, \mathbf{E}_{Igcn}^{\text{t}-1}) \tag{8}$$

where $\mathbf{E}_{time}^{\text{t}-1}$, $\mathbf{E}_{Igcn}^{\text{t}-1}$, and $\mathbf{E}_{In}^{\text{t}} \in \mathbb{R}^{(d_{fr}+d_{te})\times N}$. $\mathbf{E}_{In}^{\text{t}-1}$ represents the finally collected entity embeddings of the internal evolution encoder at the $(t\text{-}1)^{th}$ historical timestamp.

c) Internal Time Information Evolution: To completely model the temporal evolution process of a TKG, we need to consider the time information evolution process in addition to the chronological dependencies. To embed the time information of different subgraphs into the evolution procedure of the structural information, we first perform dimensionality reduction on the final internal embeddings of each timestamp:

$$\mathbf{E}_{temp}^{\text{t}} = \mathbf{W}_{time}\mathbf{E}_{In}^{\text{t}-1} + \mathbf{b}_{time} \tag{9}$$

where $\mathbf{E}_{temp}^{\text{t}} \in \mathbb{R}^{d_{fr}\times N}$ is a temporary matrix obtained after dimensionality reduction. $\mathbf{W}_{time} \in \mathbb{R}^{d_{fr}\times(d_{fr}+d_{te})}$ and $\mathbf{b}_{time} \in \mathbb{R}^{1\times N}$ are learnable parameters. Then, similar to the operation in Equation 6, we explicitly embed and update the time information during the internal structure aggregation and temporal evolution calculations:

$$\mathbf{E}_{time}^{\text{t}} = Con(\mathbf{E}_{temp}^{\text{t}}, BC(\mathbf{T}^{\text{t}})) \tag{10}$$

where $\mathbf{E}_{time}^{\text{t}} \in \mathbb{R}^{(d_{fr}+d_{te})\times N}$ is the time cell output of the t^{th} historical subgraph.

E. Decoder and Training Strategy

The time-variability problem refers to a real-world phenomenon: the historical information distributed over different timestamps plays distinct roles in forecasting future facts. To accommodate the time-variability scenarios, we use two sequences of Conv-TransE [9] models as the internal and external decoders, and adopt an online training strategy [3] to take each subgraph of the l-length historical sequence into account. In general, for a forecasting task $(s, r, ?, t\text{+}1)$ at a future timestamp t+1, the internal and external decoding processes can be represented as follows:

$$\mathbf{p}_{\text{In}}^{\text{t}+1} = \sum_{\text{T}=\text{t}-l+1}^{\text{t}} \sigma(CT_{\text{In}}^{\text{T}}(\mathbf{s}_{In}^{\text{T}}, \mathbf{r}_{In}) \cdot \mathbf{E}_{In}^{\text{T}}) \tag{11}$$

$$\mathbf{p}_{\text{Ex}}^{\text{t}+1} = \sum_{\text{T}=\text{t}-l+1}^{\text{t}} \sigma(CT_{\text{Ex}}^{\text{T}}(\mathbf{s}_{Ex}^{\text{T}}, \mathbf{r}_{Ex}) \cdot \mathbf{E}_{Ex}^{\text{T}}) \tag{12}$$

where $T \in [t-l+1, t]$ denotes the l-length historical timestamps. $\mathbf{s}_{In}^{T}$, $\mathbf{s}_{Ex}^{T}$, $\mathbf{r}_{In}$, and $\mathbf{r}_{Ex}$ denote the embeddings of the entity s and relation r from the T^{th} historical timestamp matrices $\mathbf{E}_{In}^{\text{T}}$, $\mathbf{E}_{Ex}^{\text{T}}$, $\mathbf{R}_{In}$, and $\mathbf{R}_{Ex}$, respectively. As shown in Figure 2, $CT_{\text{In}}^{\text{T}}$ and $CT_{\text{Ex}}^{\text{T}}$ denote the Conv-TransE models of the internal and external evolution decoders at the T^{th} historical timestamp, respectively. $\sigma(\cdot)$ indicates the sigmoid activation function. $\mathbf{p}_{\text{In}}^{\text{t}+1} \in \mathbb{R}^{N}$ and $\mathbf{p}_{\text{Ex}}^{\text{t}+1} \in \mathbb{R}^{N}$ are the final internal and external event forecasting scores at the $(t\text{+}1)^{th}$ future timestamp, respectively.

During online training, we simultaneously learn the roles played by internal and external evolution by summing the two scores. Finally, we adopt the cross-entropy loss for this task:

$$\mathcal{L} = -\sum_{(s,r,o,t)\in\mathcal{G}} o_t \ln(\mathbf{p}_{\text{In}}^{\text{t}+1} + \mathbf{p}_{\text{Ex}}^{\text{t}+1}) \tag{13}$$

where $\mathbf{p}_{\text{In}}^{\text{t}+1}$+$\mathbf{p}_{\text{Ex}}^{\text{t}+1} \in \mathbb{R}^{\text{N}}$. Each dimension represents the probability of forecasting the matched entity as the missing object o. o_t is the ground-truth missing entity for the forecasting task at the t^{th} timestamp.

F. Computational Complexity Analysis

To demonstrate the practical capacity of the IE-Evo model, we analyze the computational complexities of its two encoders. For the external evolution encoder, the time complexities of the external knowledge acquisition, structural aggregation, and chronological dependencies modeling processes are $O(n^2 d_{BERT})$, $O(lN)$, and $O(ld^2)$, respectively, where n is the maximum number of entity reference tokens. For the internal evolution encoder, the time complexities of the internal time information-aware aggregation, chronological dependencies modeling, and time information evolution processes are $O(lN)$, $O(ld^2)$, and $O(N)$, respectively. Therefore, the time complexity of the IE-Evo model is $O(n^2 d_{BERT}$+$l(N + d^2))$.

IV. Experiments

In this section, we empirically evaluate the performance of the IE-Evo model through multiple experiments conducted on four well-established TKG datasets.

TABLE I
PERFORMANCE (IN PERCENTAGES) ACHIEVED ON THE TKG DATASETS IN TERMS OF RAW METRICS. THE BEST AND SECOND-BEST RESULTS ARE BOLDED AND UNDERLINED, RESPECTIVELY.

Model	YAGO			ICEWS14				ICEWS05-15				ICEWS18			
	MRR	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
DistMult	44.05	49.70	59.94	20.32	6.13	27.59	46.61	19.91	5.63	27.22	47.33	13.86	5.61	15.22	31.26
ConvE	41.22	47.03	59.90	30.30	21.30	34.42	47.89	31.40	21.56	35.70	50.96	22.81	13.63	25.83	41.43
ComplEx	44.09	49.57	59.64	22.61	9.88	28.93	47.57	20.26	6.66	26.43	47.31	15.45	8.04	17.19	30.73
Conv-TransE	46.67	52.22	62.52	31.50	22.46	34.98	50.03	30.28	20.79	33.80	49.95	23.22	14.26	26.13	41.34
RotatE	42.08	46.77	59.39	25.71	16.41	29.01	45.16	19.01	10.42	21.35	36.92	14.53	6.47	15.78	31.86
R-GCN	20.25	24.01	37.30	28.03	19.42	31.95	44.83	27.13	18.83	30.41	43.16	15.05	8.13	16.49	29.00
TTransE	26.10	36.28	47.73	12.86	3.14	15.72	33.65	16.53	5.51	20.77	39.26	8.44	1.85	8.95	22.38
HyTE	14.42	39.73	46.98	16.78	2.13	24.84	43.94	16.05	6.53	20.20	34.72	7.41	3.10	7.33	16.01
TA-DistMult	44.98	50.64	61.11	26.22	16.83	29.72	45.23	27.51	17.57	31.46	47.32	16.42	8.60	18.13	32.51
RE-NET	46.81	52.71	61.93	35.77	25.99	40.10	54.87	36.86	26.24	41.85	57.60	26.17	16.43	29.89	44.37
CyGNet	46.72	52.48	61.52	34.68	25.35	38.88	53.16	35.46	25.44	40.20	54.47	24.98	15.54	28.58	43.54
xERTE	64.29	74.50	87.38	32.23	24.29	36.41	48.76	38.07	28.45	43.92	57.62	27.98	19.26	32.43	46.00
RE-GCN	63.07	71.17	82.07	41.50	30.86	46.60	62.47	46.41	35.17	52.76	67.64	30.55	20.00	34.73	51.46
TITer	<u>64.97</u>	<u>74.80</u>	<u>87.44</u>	40.90	31.77	45.84	57.67	46.62	36.46	52.29	65.23	28.44	20.06	32.07	44.33
TLogic	-	-	-	41.80	31.93	47.23	60.53	45.99	34.49	52.89	67.39	28.41	18.74	32.71	47.97
CEN	63.39	71.68	83.16	41.64	31.22	46.55	61.59	<u>49.57</u>	<u>37.86</u>	<u>56.42</u>	<u>71.32</u>	29.70	19.38	33.91	49.90
TiRGN	64.71	74.17	87.01	43.88	33.12	<u>49.48</u>	64.98	48.72	37.17	55.48	70.53	<u>32.06</u>	<u>21.08</u>	<u>36.75</u>	<u>53.62</u>
HGLS	-	-	-	**47.00**	**35.06**	-	**70.41**	46.21	35.32	-	67.12	29.32	19.21	-	49.83
CENET	55.68	62.26	76.79	35.81	26.90	39.45	53.53	39.02	28.77	43.77	58.88	26.47	17.57	29.50	44.33
PPT	-	-	-	38.42	28.94	42.50	57.01	38.85	28.57	43.35	58.63	26.63	16.94	30.64	45.43
IE-Evo	**68.43** ±0.04	**79.93** ±0.05	**89.85** ±0.08	<u>44.87</u> ±0.01	<u>34.46</u> ±0.02	**49.64** ±0.02	<u>65.26</u> ±0.01	**51.72** ±0.01	**40.03** ±0.02	**58.75** ±0.01	**72.97** ±0.03	**34.04** ±0.01	**22.86** ±0.02	**39.15** ±0.02	**55.89** ±0.03

TABLE II
DETAILS OF THE TKG DATASETS.

#Datasets	ICEWS14	ICEWS05-15	ICEWS18	YAGO
#Entities	6,869	10,094	23,033	10,623
#Relations	230	251	256	10
#Training	74,845	368,868	373,018	161,540
#Validation	8,514	46,302	45,995	19,523
#Test	7,371	46,159	49,545	20,026
#Granularity	24 hours	24 hours	24 hours	1 year

A. *Experimental Setup*

a) Datasets: Four widely-used TKG datasets are utilized for evaluation purposes, namely, YAGO [12], ICEWS05-15 [13], ICEWS14 [13], and ICEWS18 [1]. YAGO is a temporal subgraph extracted from YAGO3 [27]. ICEWS05-15, ICEWS14, and ICEWS18 are time-series political records from the Integrated Crisis Early Warning System [28]. Following previous works [1]–[4], we split the datasets in chronological order. The training set accounts for 80% of the original datasets, and the validation and test sets each account for 10% of the remaining data. Table II details the statistics of the adopted datasets.

b) Baseline Methods: The IE-Evo model is compared to multiple TKG forecasting methods, including static and dynamic. Among the static methods are ConvE [10], DistMult [8], ComplEx [7], Conv-TransE [9], R-GCN [11], and RotatE [6]. The interpolation methods include TA-DistMult [13], TTransE [14], and HyTE [12]. We focus on comparisons with some extrapolation methods, including RE-NET [1], xERTE [16], CyGNet [15], TITer [18], RE-GCN [2], TLogic [17], CEN [3], TiRGN [4], CENET [22], HGLS [5], and PPT [23]. We detail the baseline methods in Section II.

c) Evaluation Metrics: For the TKG forecasting task, four widely adopted metrics are used to evaluate the performance of the tested models. The MRR represents the average rankings of missing entities. Hits@1/3/10 indicate the proportions of queries with missing entities in the top 1/3/10 forecasting results. Following RE-GCN [2], the mean results of object forecasting and subject forecasting tasks are reported, and for the YAGO dataset, only the Hits@3, Hits@10, and MRR metrics are reported. According to previous works, the static filtering setting is not suitable for temporal scenarios. In addition, the filtering operation tends to obtain better results by filtering out conflicting candidates and cannot reflect the real abilities of models. Thus, we adopt the raw setting and report the raw results obtained by the tested methods.

d) Implementation Details: We implement and train the IE-Evo model using PyTorch and a GeForce RTX 3090 GPU. We configure the parameters on the validation set during the traditional training process and fine-tune them according to the newly emerging historical information during the validation and test processes. We set the batch size to match the size of each timestamp, and use 100 training epochs across all datasets to ensure the convergence of the model. We set the historical length l to 4 for the ICEWS18 dataset, 9 for the ICEWS05-15 dataset, and 3 for the YAGO and ICEWS14 datasets. The historical length l is analyzed in detail in Section IV-G. For the BERT model, we set the dropout rate to 0.1, the hidden size d_{BERT} to 768, the number of hidden layers to 12, and the number of attention heads to 12. For the R-GCN models, the number of hidden layers and the dropout rate are set to 2 and 0.2, respectively. For the Conv-TransE models, we set the dropout rate to 0.2, the kernel size to 2×3, and the number of kernels to 50. We set d_{fr} = 96 and d_{lm} = d_{te} = 32 for the YAGO dataset, and d_{fr} = d_{lm} = d_{te} = 64 for the ICEWS18, ICEWS05-15, and ICEWS14 datasets. In addition, the Adam optimizer is chosen for training, and the learning rate is set to 0.001. Following TiRGN [4] and RE-GCN [2], we also add

TABLE III
ABLATION STUDY RESULTS (IN PERCENTAGES) OBTAINED ON ALL THE DATASETS.

Datasets	YAGO	ICEWS14	ICEWS05-15	ICEWS18
Internal **Evo**lution	60.43	12.00	14.95	10.41
External **Evo**lution	67.23	43.58	49.46	30.00
IE-Evo	**68.43**	**44.87**	**51.72**	**34.04**

static graph constraints for the ICEWS series datasets.

For the static methods, the time dimension is simply removed. Following the same dataset and splitting strategy, some of the baseline results are taken from [2]. For the important baseline methods including TITer [18], CEN [3], xERTE [16], TLogic [17], TiRGN [4], and CENET [22], we replicate the results with their public codes and default parameters under the raw setting. For the HGLS [5] and PPT [23] models without open-source codes, we adopt their reported results due to the use of the same evaluation metrics and settings.

B. Forecasting Results Obtained on TKGs

In this section, the capability of the IE-Evo model is compared with that of multiple dynamic and static methods.

As Table I shows, the IE-Evo model significantly outperforms the static methods because they do not take the time dimension in which the facts occur into account. Some interpolation methods, like TTransE and HyTE, prioritize the embeddings of time information, but overlook the evolution of historical sequences. Excluding the unsuitability of forecasting scenarios, this may result in even worse performance than that of some static methods. The advanced extrapolation methods mainly focus on capturing the structural evolutionary patterns while ignoring the time information embeddings and their evolution. For example, RE-NET, RE-GCN, and CEN all regard a TKG as a sequential stack of subgraphs. Under such circumstances, distinct time information is not available, and only the chronological dependencies of temporal evolution can be considered. TLogic treats a TKG as a rule learner, with fixed rule templates that provide substantial results; however, it simultaneously ignores the structural aggregation and temporal evolution processes within a TKG. CyGNet only considers the timestamp information where the forecasting facts are located, but the time information and its evolution between the historical subgraphs are still ignored. TiRGN and HGLS both attempt to learn the embeddings of the timestamp information but neglect its evolution. IE-Evo outperforms the dynamic methods because it simultaneously considers the historical timestamp embeddings and evolution patterns, and thus has the ability to model the time information evolution process on the basis of chronological dependencies. We note that IE-Evo performs slightly worse than HGLS on the ICEWS14 dataset, because HGLS simply adds explicit edges to connect the concurrent entities at different timestamps in the global KG. This allows HGLS to reap larger gains on some small-scale graphs (e.g., ICEWS14), which contain many sparse facts without complex semantics. However, as the results show, IE-

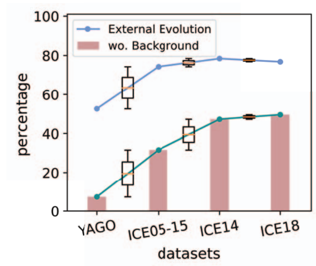

Fig. 3. Study on the external evolution process and background information in terms of all the datasets.

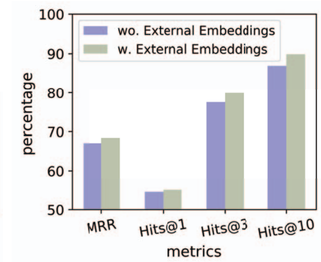

Fig. 4. Study on the role of external embeddings in the external evolution process in terms of the YAGO dataset.

Evo possesses stronger robustness than HGLS to accommodate the modeling of large-scale multi-relational graphs (e.g., ICEWS05-15 and ICEWS18).

On the other hand, constrained by the lengths of the given subgraph sequences, limited background information is available in the historical space (the background knowledge base). Specifically, TITer, xERTE, and CENET attempt to address the challenges faced by the encoder in terms of the lack of background information for some forecasting facts within a TKG. Among them, TITer and xERTE aggregate and induce the representations of entities with scarce or even no historical information through neighborhood information; nevertheless, as shown in Table I, their commitment to mining the internal evolution process has little effect on the results. CENET attempts to obtain the representations of non-historical entities via contrastive learning; however, it restricts historical entities to only the one-hop entities associated with the query facts, and instead fails to effectively model most historical entities. CENET [22] reports results under an unreasonable static filtering setting, but when a more reasonable raw setting is used for evaluation, as shown in Table I, it performs catastrophically. The IE-Evo model performs better than these methods because we introduce much semantically rich background information from an external PLM and model the evolution process of external knowledge for addressing the semantically poor entities within a TKG. Each entity has its own textual references in the real world; for example, the 1^{st} entity corresponds to North Korea in the ICEWS18 dataset. For entities with sufficient internal background information, the addition of external knowledge can enhance their representations. Thus, through the abovementioned two aspects, IE-Evo outperforms the baseline models in terms of nearly all evaluation metrics.

C. Ablation Study

In this section, we conduct an ablation study on all the datasets and present the results using the representative MRR metric due to space constraints.

As Table III shows, we use the scores of the internal and external decoders to predict future events to express the roles of internal and external evolution in the IE-Evo model. It is observed that the performance achieved using any single module is lower than that of the complete IE-Evo model, which proves that the internal and external modules are

TABLE IV
STUDY ON THE DIFFERENT EXTERNAL PLMS IN TERMS OF THE ICEWS18 DATASET.

PLMs	Params	MRR	Hits@1	Hits@3	Hits@10
bert-base-uncased	109.48 M	34.04	22.86	39.15	55.89
bert-base-cased	108.31 M	33.99	22.78	39.10	55.80
bert-large-uncased	333.58 M	33.98	22.77	39.13	55.80
roberta-base	124.65 M	34.03	22.83	39.11	55.89
roberta-large	355.36 M	34.01	22.81	39.12	55.93

both helpful for performing event forecasting. Nevertheless, in general, the external evolution performance is better than that achieved when confined to an internal TKG. This proves the value of the background information introduced by external knowledge. We find that the internal evolution processes of the ICEWS series datasets perform much worse than that of the YAGO dataset, which means that the external evolution process plays a greater role in the final forecast. Therefore, we continue to explore the relationship between external evolution and background information.

As shown in Figure 3, we count and compare the proportions of facts facing a lack of background information in each dataset versus the roles played by the corresponding external evolution processes. If a forecasting fact of the test sets does not reappear in history, it cannot obtain any background information. We express the contribution weight of the external evolution module as the ratio of the MRR of external evolution to the sum of the MRRs of the internal evolution and external evolution modules. Then, a positive correlation can be observed between the abovementioned two curves. The degree of the lack of background information faced by the ICEWS series datasets is larger than that of the YAGO dataset, and thus the external evolution module plays a more important role in these datasets. Figure 3 also demonstrates the success and effectiveness achieved by IE-Evo through the introduction of external background information.

D. Study on External Knowledge

Figure 3 shows the importance of external evolution. In this section, we further perform a study on the external embeddings in the external evolution process.

Because the static constraints of the ICEWS series datasets preserve the learned parameters and are hard to decouple, we perform this experiment on the YAGO dataset. In our proposed IE-Evo model, external embeddings are introduced by external knowledge via a PLM as the inputs of the external GCNs. We use or exclude the external embeddings in the external evolution process and report the metric comparison results to study the functionality of the introduced external embeddings. To exclude the external embeddings from the external evolution module, we use random initialization and retain the normal distribution according to the front-end embeddings, thus masking the external knowledge. As shown in Figure 4, we report the comparisons among the Hits@10, Hits@3, Hits@1, and MRR metrics yielded with and without external embeddings. It can be observed that the IE-Evo model

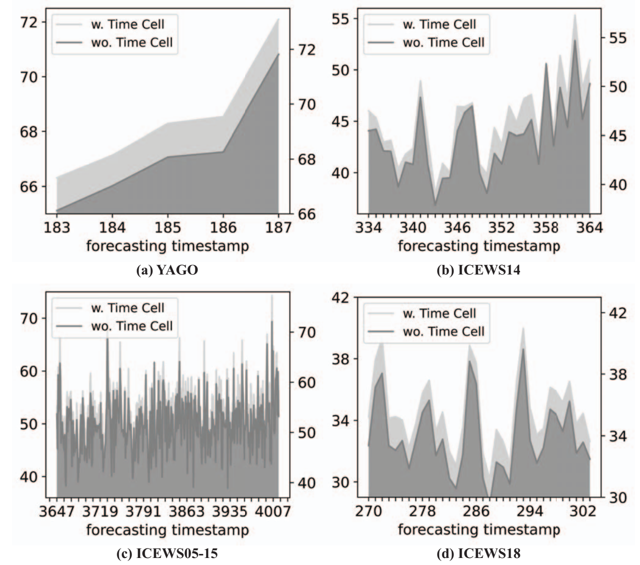

Fig. 5. Study on the role of time information evolution in terms of all the datasets. The left y-axis and right y-axis show the performance (MRR %) achieved by IE-Evo with and without the time cells, respectively.

performs better with the addition of external knowledge in terms of all the metrics. This demonstrates that the external embeddings improve the performance of the external evolution module and the entire model.

E. Study on Different External PLMs

In this section, we continue to study the model performance achieved on the ICEWS18 dataset after providing different external embeddings through different external PLMs.

As Table IV shows, in addition to the bert-base-uncased model, we also attempt other pre-trained language models (including RoBERTa [29] and BERT series models): roberta-base [2], roberta-large [3], bert-base-cased[4], and bert-large-uncased [5]. We input the same real-world textual references of each entity into different PLMs and then obtain different external embeddings for the corresponding entity.

For the experimental PLMs, we use their default parameter configurations. We also report the parameter scales of different PLMs, as shown in Table IV. We find that the experimental results are similar across different PLMs, despite the varying spatial complexities. Through these results, it becomes evident that our proposed IE-Evo model exhibits remarkable autonomy, as it does not rely on any particular pre-trained language model. This characteristic highlights its robust ability to adapt and perform effectively across diverse scenarios, underscoring its strong generalization capability.

F. Study on the Evolution of Time Information

In this section, we study the functionality of the time information evolution process in terms of all the datasets.

[2]https://huggingface.co/roberta-base
[3]https://huggingface.co/roberta-large
[4]https://huggingface.co/bert-base-cased
[5]https://huggingface.co/bert-large-uncased

Because studies on temporal evolution (including chronological dependencies and time information evolution) are strongly related to the continuous development of timestamps, we perform experiments on the consecutive timestamps of different datasets. As Figure 5 shows, the x-axes of different subplots denote the forecasting timestamps, where each number denotes a day for the ICEWS series datasets and a year for the YAGO dataset. In addition, the experimental time period ranges from the 183^{rd} to the 187^{th} forecasting timestamps of the YAGO dataset, from the 334^{th} to the 364^{th} forecasting timestamps of the ICEWS14 dataset, from the 3647^{th} to the 4016^{th} forecasting timestamps of the ICEWS05-15 dataset, and from the 270^{th} to the 303^{rd} forecasting timestamps of the ICEWS18 dataset. The y-axis represents the MRR metric results. In our proposed IE-Evo model, we use an RNN and design a time cell to capture the chronological dependencies and time information evolution, respectively, within the internal evolution module. Thus, we compare the model performances achieved with and without the time cells at different forecasting timestamps to study the role played by time information evolution.

Specifically, to remove the time cells from the internal evolution module, we randomly initialize the embeddings of the first historical timestamp with the same normal distribution as that of the front-end embeddings. Then, we remove the time cells at the subsequent historical timestamps and directly feed the output of the RNN from the previous timestamp as the input to the GCN at the subsequent timestamp. Thus, only the chronological dependencies of the temporal evolution process can be modeled. As Figure 5 shows, the light gray areas represent the complete IE-Evo model, which incorporates the embeddings of time information through time cells during the process of temporal evolution, while the dark gray areas represent the MRR results obtained without the time cells. IE-Evo performs better at any single timestamp during the evolution of time when considering the representation and evolution of time information. This demonstrates that explicitly considering the embeddings and evolution of time information can help the forecasting facts to fully distinguish and utilize the information obtained at different historical timestamps.

G. Study on Different Historical Lengths

In this section, we study the influence of different historical sequence lengths (corresponding to the hyperparameter l) on model performance.

As shown in Figure 6, we compare and choose the lengths of historical subgraph sequence l from $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ according to IE-Evo's performances of different datasets. More specifically, we label the values of historical sequence length at the peak MRR and Hits@1 metrics for various datasets.

It is observed that IE-Evo achieves the best performance on the YAGO, ICEWS14, ICEWS05-15, and ICEWS18 datasets at historical lengths of 3, 3, 9, and 4, respectively, demonstrating that short-term historical information is sufficient to support effective prediction. Furthermore, IE-Evo can maintain relatively stable performance as the length of the historical sequence increases without being limited to the effects of long-term noise information. We note that the performance of IE-Evo degrades significantly with increasing historical scope on the ICEWS14 dataset. This may be due to the small factual size of the ICEWS14 dataset, which leads to uncertainty in the model performance when increasing the length of the graph sequence modeling. In summary, IE-Evo possesses certain stability as the historical length increases, and this capability also corresponds to its advantage of explicitly modeling the time information evolution in Section IV-F.

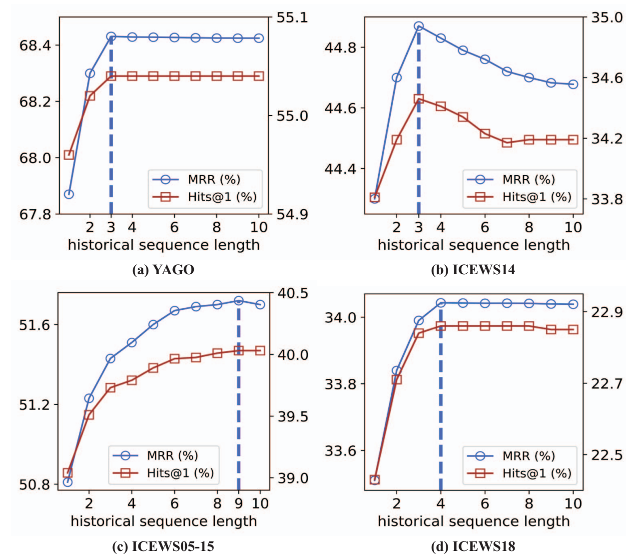

Fig. 6. Study on the model performances achieved with different historical lengths in terms of all the datasets. The left y-axis and right y-axis show the MRR and Hits@1 values, respectively.

V. Conclusions

This paper proposes IE-Evo to address the lack of available background information and the evolution of time information in TKG forecasting. IE-Evo introduces external background information via a PLM and models the evolution of external knowledge. In addition, it comprehensively models the time information-aware structural aggregation, the chronological dependencies, and the time information evolution processes within a TKG through an internal evolution encoder. By seamlessly incorporating knowledge from inside and outside TKGs to enhance the representations of entities, IE-Evo achieves remarkable improvements over the baseline models.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China under Grants No.62072203, and Australian Research Council Under Grants DP22010371, LE220100078.

References

[1] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 6669–6683. ACL, 2020.

[2] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 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, Virtual Event, Canada, July 11-15, 2021*, pages 408–417. ACM, 2021.

[3] Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. Complex evolutional pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 290–296. ACL, 2022.

[4] Yujia Li, Shiliang Sun, and Jing Zhao. 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, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 2152–2158. ijcai.org, 2022.

[5] Mengqi Zhang, Yuwei Xia, Qiang Liu, Shu Wu, and Liang Wang. Learning long- and short-term representations for temporal knowledge graph reasoning. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 2412–2422. ACM, 2023.

[6] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[7] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016.

[8] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[9] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3060–3067. AAAI Press, 2019.

[10] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press, 2018.

[11] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of the 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018.

[12] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2001–2011. ACL, 2018.

[13] Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821. ACL, 2018.

[14] Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. Towards time-aware knowledge graph completion. In *Proceedings of the 26th International Conference on Computational Linguistics, December 11-16, 2016, Osaka, Japan*, pages 1715–1724. ACL, 2016.

[15] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021*, pages 4732–4740. AAAI Press, 2021.

[16] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[17] Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*, pages 4120–4127. AAAI Press, 2022.

[18] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8306–8319. ACL, 2021.

[19] Kangzheng Liu, Feng Zhao, Hongxu Chen, Yicong Li, Guandong Xu, and Hai Jin. Da-net: Distributed attention network for temporal knowledge graph reasoning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1289–1298. ACM, 2022.

[20] Kangzheng Liu, Feng Zhao, Guandong Xu, Xianzhi Wang, and Hai Jin. Temporal knowledge graph reasoning via time-distributed representation learning. In *IEEE International Conference on Data Mining, ICDM 2022, Orlando, FL, USA, November 28 - Dec. 1, 2022*, pages 279–288. IEEE, 2022.

[21] Kangzheng Liu, Feng Zhao, Guandong Xu, Xianzhi Wang, and Hai Jin. RETIA: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 1761–1774. IEEE, 2023.

[22] Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. Temporal knowledge graph reasoning with historical contrastive learning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, February 7 - 14, 2023*, page to be appeared. AAAI Press, 2023.

[23] Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. Pre-trained language model with prompts for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, page to be appeared. ACL, 2023.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. ACL, 2019.

[25] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[27] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *Proceedings of the Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org, 2015.

[28] Elizabeth Boschee, Jennifer Lautenschlager, Sean OBrien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data. *Harvard Dataverse*, 12, 2015.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.