SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2025, Vol. 68, Iss. 2, 122104:1–122104:17
https://doi.org/10.1007/s11432-023-4073-y

RE-SEGNN: recurrent semantic evidence-aware graph neural network for temporal knowledge graph forecasting

Wenyu CAI†, Mengfan LI†, Xuanhua SHI*, Yuanxin FAN, Quntao ZHU & Hai JIN

National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Received 1 July 2023/Revised 1 November 2023/Accepted 8 March 2024/Published online 8 January 2025

Abstract Temporal knowledge graph (TKG) reasoning, has seen widespread use for modeling real-world events, particularly in extrapolation settings. Nevertheless, most previous studies are embedded models, which require both entity and relation embedding to make predictions, ignoring the semantic correlations among different entities and relations within the same timestamp. This can lead to random and nonsensical predictions when unseen entities or relations occur. Furthermore, many existing models exhibit limitations in handling highly correlated historical facts with extensive temporal depth. They often either overlook such facts or overly accentuate the relationships between recurring past occurrences and their current counterparts. Due to the dynamic nature of TKG, effectively capturing the evolving semantics between different timestamps can be challenging. To address these shortcomings, we propose the recurrent semantic evidenceaware graph neural network (RE-SEGNN), a novel graph neural network that can learn the semantics of entities and relations simultaneously. For the former challenge, our model can predict a possible answer to missing quadruples based on semantics when facing unseen entities or relations. For the latter problem, based on an obvious established force, both the recency and frequency of semantic history tend to confer a higher reference value for the current. We use the Hawkes process to compute the semantic trend, which allows the semantics of recent facts to gain more attention than those of distant facts. Experimental results show that RE-SEGNN outperforms all SOTA models in entity prediction on 6 widely used datasets, and 5 datasets in relation prediction. Furthermore, the case study shows how our model can deal with unseen entities and relations.

Keywords knowledge graph reasoning, temporal knowledge graph, Hawkes process, semantic evidence

Citation Cai W Y, Li M F, Shi X H, et al. RE-SEGNN: recurrent semantic evidence-aware graph neural network for temporal knowledge graph forecasting. Sci China Inf Sci, 2025, 68(2): 122104, https://doi.org/10.1007/s11432-023-4073-y

1 Introduction

As a means of representing factual information through a graph structure, knowledge graphs (KGs) have been applied in numerous scenarios. The knowledge graph stores knowledge in the form of a multi-edge directed graph, and each fact in KG is represented by a triple (e_s, r, e_o), where e_s, e_o represent the subject entity and the object entity, respectively, and r represents the relation. However, many facts have dynamic features, which means that some events occur only during a certain period or even at a specific timestamp. To solve the problem that static KGs cannot model the temporal features of facts, the temporal knowledge graph (TKG) is developed. TKG adds a time dimension to the static knowledge graph, and expands the triple to a quadruple (e_s, r, e_o, t), where t represents a timestamp.

TKG reasoning is mainly classified into two categories: interpolation and extrapolation. For a given TKG from T_0 to T_t, under the interpolation setting, the timestamp t of the fact to predict satisfies $T_0 < t < T_t$. On the contrary, in the extrapolation scenario, the timestamp t of the fact to predict satisfies $t > T_t$, the subject entity or relation may never have been seen before. Extrapolation setting is currently gaining more attention because it is more in line with modeling the real world, where we cannot know what will happen at future timestamps.

* Corresponding author (email: xhshi@hust.edu.cn)
† These authors contributed equally to this work.

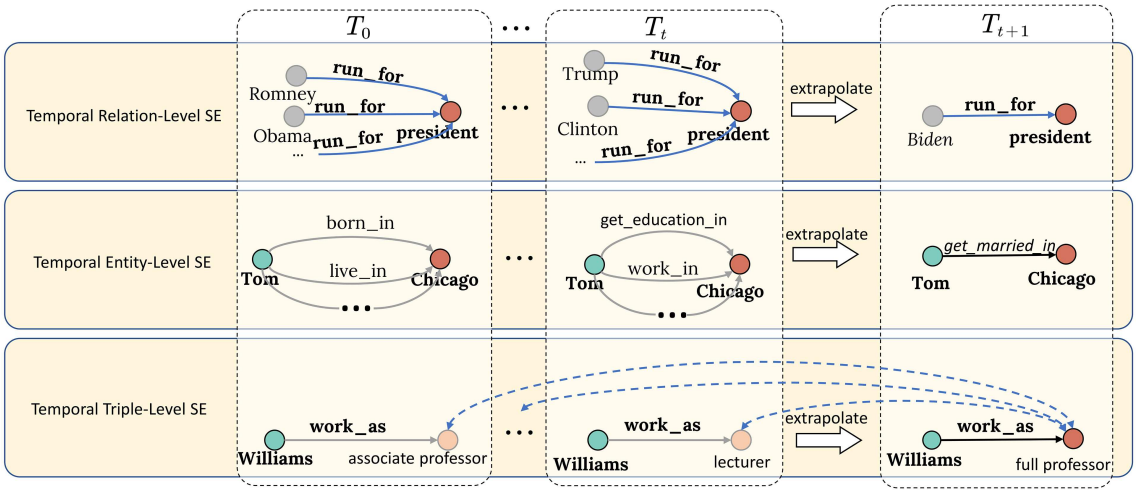

Figure 1 (Color online) Examples of the three levels of TSE. When performing prediction at time T_{t+1}, the unseen entity or relation is marked in italics. For relation-level SE, Biden has never been seen before T_{t+1}, but when facing query (Biden, run_for, ?, T_{t+1}), we can still make prediction by learning relation-level SE between relation run_for and entity president in history, to know the most likely answer is president. Similarly, for entity-level, when facing unseen relation get_married_in for query (Tom, get_married_in, ?, T_{t+1}), we can learn entity-level SE between Tom and Chicago, to make prediction that the answer is Chicago. As for triple-level SE, it is used to learn the semantics of triple history; for example, we know (Williams, work_as, Assistant Professor, T_0) and (Williams, work_as, lecturer, T_t), the answer of query (Williams, work_as, ?, T_{t+1}) is very likely to be a full professor because full professor is semantically similar to associate professor and lecturer.

Most previous extrapolation work faced two challenges: On one hand, most models are embedding-based models. When facing queries containing unseen entities or relations, the embedding of these unseen entities or relations has not been learned because there are no facts in the history that contain those unseen entities or relations. This leads to randomness in the predictions.

On the other hand, many methods cannot consider both the time difference and frequency of historical facts. (1) Some models [1–3] use an auto-regressive recurrent neural network (RNN) to learn from the past event sequences, which causes the model forget long historical sequences. (2) Other models [4,5] tend to capture repetitive history, but the time-variant features of entities' behavior are ignored. Thus, facts with high frequency tend to be predicted, but this is not always correct because entities' behavior may change over time. Take "Susan walked home after work for years then drove home for months, but she walked home yesterday" as an example, regarding the question of "How Susan will go home tomorrow?", the former model tends to answer "walking" because walking home is the most recent event (yesterday), and the latter model also tends to answer "walking" because walking home has the highest frequency. However, considering all the events that have occurred in the past month, we will find that the frequency of driving is much greater than walking, so the correct answer may be to drive. This example illustrates that accounting for both the time difference and frequency of past events is necessary to obtain correct predictions.

We argue that the aforementioned two challenges stem from the failure to adequately consider semantic evidence in both spatial and temporal dimensions. To this end, we propose recurrent semantic evidence-aware graph neural network (RE-SEGNN), which can use semantic evidence to solve the above problems like human thinking. Semantic evidence refers to the evidence for inference that is found by looking for connections between entities and entities, entities and relations, or entity-relation-entity triples. For the former problem, we learn the dynamic embeddings not only from triples but also from the co-occurrence pairs between entities and relations. From a spatial perspective, we model temporal semantic evidence (TSE) at three levels at each timestamp to reveal temporal semantic information. (1) Temporal relation-level semantic evidence indicates the semantic evidence of co-occurrence between the relation and the object entity. (2) Temporal entity-level semantic evidence represents the associative semantics between the subject and object entity. (3) Temporal triple-level semantic evidence indicates the similarity of historically similar facts. Some cases of the three levels of SE are shown in Figure 1.

For the latter problem, we learn repetitive and temporal evolving semantics over timestamps to help the TSE learn more information from history. Similar to the TSE, we consider not only the available history at the triple level but also the changing trend of change at the relation and entity level. We call these three types of semantic evidence as historical triple-level SE, historical relation-level SE, and

historical entity-level SE, collectively referred to as the historical semantic evidence (HSE).

To model the changing historical effect at the current timestamp, we introduce the Hawkes process [6] to consider the number of times a fact occurred, as well as the interval between the time of related historical facts and the current time. The Hawkes process is a class of stochastic process with the self-exciting characteristic, which means that the occurrence of an event can increase the probability of similar events in future timestamps, and the influence decays as the time difference increases. Specifically, we utilize a time-difference-dependent function to quantify the influence exerted by the TSE at each past timestamp on the current timestamp. The effective accumulation of this function yields the total influence from all past timestamps, which we refer to as the HSE. To effectively capture this information, we facilitate an interaction between HSE and TSE at the current timestamp, yielding a rational effect through which the HSE from past timestamps influences the current timestamp.

Overall, our model makes the following three main contributions.

(1) We propose the RE-SEGNN[1)], which leverages semantic evidence to handle unseen entities and relations in TKG, while efficiently perceives the impact of historical facts on the current timestamp. To our knowledge, we are the first to exploit semantic evidence for TKG.

(2) To obtain time-aware semantic information of TKG when facing unseen entities or relations, we propose two types of semantic evidence: TSE and HSE. TSE aims to learn the time-aware semantic dependencies among the evolving structures of TKG. HSE aims to incorporate the time-evolving features of historical events and can be obtained by applying the Hawkes process to extract the historical semantics.

(3) Experimental results on six widely used benchmarks show that RE-SEGNN outperforms all state-of-the-art (SOTA) models on entity prediction and achieves SOTA performance on relation prediction, and case studies confirm that the TSE and HSE can enhance the extrapolation ability of RE-SEGNN when dealing with unseen entities or relations.

The rest of this paper is organized as follows. Section 2 shows some related work. Section 3 first formally defines TKG and TKG reasoning with extrapolation settings including symbolic definitions, and then introduces the details of each component of RE-SEGNN. In Section 4, we introduce the details of the experiment and compare RE-SEGNN with other relevant SOTA models, and we then conduct an ablation study to show the effectiveness of RE-SEGNN. Finally, Section 5 summarizes the conclusion.

2 Related work

In this section, we review the related work on static KG reasoning, and then present TKG reasoning under the interpolation and extrapolation settings.

2.1 Static knowledge graph reasoning

The methods of static KG reasoning mainly include embedding-based models, deep learning-based models, and path-based models. The distance model is a typical example of embedding-based reasoning, in which the idea is to minimize the distance between related entities in the embedding space. TransE [7], TransH [8], and other distance models aim to minimize the distance between related entities in different spaces. Another embedding-based model is the tensor decomposition-based model, which constructs a third-order tensor and decomposes the tensor into low-dimensional vector space to get node embedding. The earliest proposed model of this type is RESCAL [9]. DistMult [10] and ComplEx [11] are the success work of RESCAL. Deep learning-based reasoning model uses deep neural networks to get a prediction. For example, ConvE [12] uses a 2D-convolutional network to model triples. Similar studies include CapsE [13] and ConvKB [14]. Path-based reasoning mostly uses reinforcement learning to learn path information, and obtains one-hop or multi-hop path information of the graph. Path-based reasoning generally has good explainability. DeepPath [15] is also outstanding work in this field.

In recent years, some studies have been dedicated to exploring the integration of semantic information into the structural details of KGs. However, the semantic information referred to in these studies primarily pertains to the type information of entities or relations, or natural language text descriptions of entities or relations. KG-BERT [16] pioneered the incorporation of natural language text descriptions for entities and relations into various tasks of knowledge graphs. By utilizing the BERT [17] model and a cross-encoder architecture, KG-BERT achieved a straightforward fusion of semantic information.

1) Our source code is available at https://github.com/CGCL-codes/resegnn.

Subsequent studies have similarly focused on the integration of text descriptions with KGs; for instance, SimKGC [18], adopting a bi-encoder architecture, demonstrated commendable performance in the task of KG completion. LASS [19], on the other hand, fine-tuned pre-trained language models based on natural language text using a structure-based loss function, enhancing the performance in node classification and link prediction. Furthermore, some studies attempt to leverage the type information of entities and relations as supplementary information. TEMP [20] utilizes the type information of entities to improve the performance on multi-hop logical queries. AutoETER [21], integrating hierarchical type triplets encoding, proficiently accomplished the task of KG completion.

As shown in Figure 1, our work defines semantic evidence as statistical patterns within datasets, that is organized into three levels: relation, entity, and triple. Such semantics unfold across spatial and temporal dimensions in TKG. The most prominent difference between the semantic evidence referenced in our model and the semantic information utilized by previous models lies in whether the information is present within the KG itself. Moreover, our approach pioneers in the domain of TKG reasoning by extending semantic evidence to TKG, seamlessly amalgamating it with the Hawkes process. This novel integration facilitates a detailed and effective perception of semantic evidence from past timestamps. Moreover, our model excavates the semantic evidence inherent in the KG at each timestamp and considers the temporal evolution of this evidence, yielding innovations in the domain of TKG reasoning by extending semantic evidence to TKG and seamlessly integrating it with Hawkes process modeling. This novel integration facilitates detailed and effective perception of semantic evidence from past timestamps. Remarkably, our model achieves SOTA performance without relying on any additional information.

2.2 Temporal knowledge graph reasoning

Under the interpolation setting, many TKG reasoning models extend static KG models by adding a time factor. TTransE [22] extends the static method TransE [7] by adding a time dimension so that the distance model can be used on the TKG. Following transH [8], HyTE [23] projects the facts at different timestamps onto the corresponding time hyperplanes to embed entities and relations. A series of tensor decomposition models such as TA-DistMult [24] and TNTComplEx [25], add a time dimension to DistMult [10] and ComplEx [11], in this case, the third-order tensor decomposition in static KG is changed to fourth-order, which can represent temporal characteristics very well.

Under the extrapolation setting, Know-Evolve [26] uses a temporal point process to consider the frequency of events to obtain graph representations, CyGNet [4] uses a copy-generation mechanism to consider repeated history for prediction, and RE-NET [1] uses an RNN in the time window to learn historical facts. To predict future events, TiTer [27] uses a temporal path-based reinforcement learning method, and CluSTeR [28] uses a similar method to capture clues, making its prediction results interpretable. Similarly, xERTE [29] utilizes a sequential reasoning process on subgraph, making the inference results explainable. rGAlt [30] uses a transformer to mine the fact precursors of TKG to predict facts. RE-GCN [2] learns structural dependencies through a relational-aware graph convolution network (RGCN) at each timestamp and then updates alongside different timestamps to obtain the representation. TiRGN [31] introduces a local-global structure to consider history and structure at the same time so that it can better predict repeated historical events. RETIA [32] uses hyperrelation embeddings to get aggregation of entities and relations. TECHS [33] incorporates both graph structures and topological structures to make predictions, RPC [34] uses relational and periodic correspondence units to learn representations of entities and relations. GHT [35] employs two variants of the Transformer to capture the structural knowledge and temporal evolution information at each moment, respectively, while simultaneously utilizing the Hawkes process to further enhance the simulation of the feature variation over time. However, capturing all historical information is bound to increase the model's complexity, and may also lead to the neglect of key information. Our method merely perceives the semantic evidence from historical timestamps, managing to save on training time overhead without sacrificing model accuracy. Some studies [36–38] attempted to introduce neural networks to parameterize the Hawkes process, aiming to enhance its effectiveness in perceiving the evolution of features over time during the training process. However, these studies do not model structure information.

3 Methodology

In this section, we first define our TKG reasoning task with extrapolation settings. Next, we introduce how the TSE encoder and HSE encoder learn TSE and HSE, respectively. Then, we show how the SE-

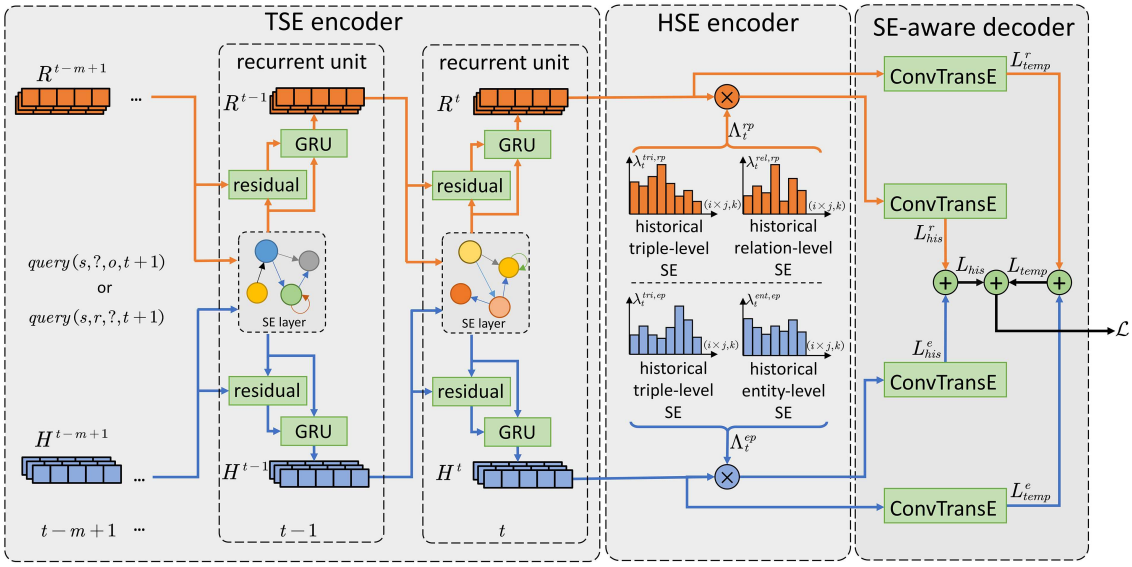

Figure 2 (Color online) RE-SEGNN architecture. RE-SEGNN contains two encoders: the TSE encoder and the HSE encoder, which can capture the TSE and HSE, respectively. RE-SEGNN also contains a decoder that is used to obtain scores of candidate entities via ConvTransE [39].

aware decoder combines the two types of SE to obtain the prediction scores. Finally, we show how our model learns parameters to make predictions. The architecture of our model is shown in Figure 2 [39].

3.1 Notations

A TKG can be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}$ represent the sets of entities, relations, timestamps, and facts, respectively. Each fact can be written as a quadruple (e_s, r, e_o, t), where $e_s, e_o \in \mathcal{E}$ denotes the subject entity and object entity, $r \in \mathcal{R}$ denotes the relation, $t \in \mathcal{T}$ denotes the timestamp of the fact, and the subgraph of the TKG at timestamp t is defined as $\mathcal{G}_t$, which can be viewed as a multi-edge directed graph. Therefore, a TKG can be represented as a sequence of multi-edge directed graphs with different timestamps.

The TKG entity prediction task can be conceptualized as a completion task for missing quadruples of the form $(e_s, r, ?, t_q)$ or $(?, r, e_o, t_q)$, while the relation prediction task can be regarded as the completion of missing quadruples of the form $(e_s, ?, e_o, t_q)$. For a TKG with timestamps ranging from 0 to T, since the purpose of the extrapolation setting is to predict future events, t_q satisfies $t_q > T$. If the entity in the query does not appear before timestamp T, then we call it an unseen entity, and there can be unseen relations as well. To simplify the task, we introduce the concept of the inverse relation, that is, for each fact $(e_s, r, e_o, t) \in \mathcal{F}$, we introduce an inverse relation (e_o, r^{-1}, e_s, t) and add it to $\mathcal{F}$, so the subject entity prediction $(?, r, e_o, t_q)$ can be regarded as the corresponding tail entity prediction $(e_o, r^{-1}, ?, t_q)$.

3.2 Temporal semantic evidence encoder

For a TKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, $\mathcal{T}$ is $[0, T]$. The TSE encoder considers the events with the latest m timestamps $[T-m+1, T]$ to learn the structural dependencies.

Time-aware semantic evidence layer. At time t, the entity matrix $\boldsymbol{H}_{t-1} \in \mathbb{R}^{|\mathcal{E}| \times d}$ and relation matrix $\boldsymbol{R}_{t-1} \in \mathbb{R}^{|\mathcal{R}| \times d}$ from time $t-1$ are used as the input of SE layer, where d represents the embedding's dimension. We use one-layer SEGNN [40] to learn entities' embedding, and the temporal relation-level SE can be obtained by

$$\boldsymbol{h}_{i,t}^{\mathrm{rel}} = \sigma\left(\sum_{r_j \in \mathcal{N}_{i,t}^{\mathrm{rel}}} \alpha_{ij}^{\mathrm{rel}} \boldsymbol{W}^{\mathrm{rel}} \boldsymbol{r}_{j,t}\right), \alpha_{ij}^{\mathrm{rel}} = \frac{\exp(\boldsymbol{r}_{j,t}^{\mathrm{T}} \boldsymbol{h}_{i,t})}{\sum_{r_k \in \mathcal{N}_{i,t}^{\mathrm{rel}}} \exp(\boldsymbol{r}_{k,t}^{\mathrm{T}} \boldsymbol{h}_{i,t})}, \tag{1}$$

where $\boldsymbol{h}_{i,t}^{\mathrm{rel}} \in \mathbb{R}^d$ represents the relation-level SE of the entity e_i at timestamp t, $\boldsymbol{r}_{j,t}$ represents the embedding of relation r_j at timestamp t, and $\mathcal{N}_{i,t}^{\mathrm{rel}}$ represents the connected relation of entity e_i at

timestamp t. We only aggregate the semantic information based on facts that occur at time t, so $\mathcal{N}_{i,t}^{\text{rel}} = \{r_j \mid (e_j, r_j, e_i, t) \in \mathcal{F}_t\}$, $\boldsymbol{W}^{\text{rel}} \in \mathbb{R}^{d\times d}$ represents a linear matrix, σ is the activation function, and α_{ij}^{rel} denotes the attention between entity i and relation j. Similarly, we obtain the entity-level SE and triple-level SE of e_i by

$$\boldsymbol{h}_{i,t}^{\text{ent}} = \sigma\left(\sum_{e_j\in\mathcal{N}_{i,t}^{\text{ent}}} \alpha_{ij}^{\text{ent}} \boldsymbol{W}^{\text{ent}} \boldsymbol{h}_{j,t}\right), \alpha_{ij}^{\text{ent}} = \frac{\exp(\boldsymbol{h}_{j,t}^{\text{T}} \boldsymbol{h}_{i,t})}{\sum_{e_k\in\mathcal{N}_{i,t}^{\text{ent}}} \exp(\boldsymbol{h}_{k,t}^{\text{T}} \boldsymbol{h}_{i,t})}, \tag{2}$$

$$\boldsymbol{h}_{i,t}^{\text{tri}} = \sigma\left(\sum_{(e_j,r_j)\in\mathcal{N}_{i,t}^{\text{tri}}} \alpha_{ij}^{\text{tri}} \boldsymbol{W}^{\text{tri}} \varphi_{jj}\right), \alpha_{ij}^{\text{tri}} = \frac{\exp\left(\varphi_{jj}^{\text{T}} \boldsymbol{h}_{i,t}\right)}{\sum_{(e_k,r_k)\in\mathcal{N}_{i,t}^{\text{tri}}} \exp\left(\varphi_{kk}^{\text{T}} \boldsymbol{h}_{i,t}\right)}, \tag{3}$$

where $\mathcal{N}_{i,t}^{\text{ent}} = \{e_j \mid (e_j, r_j, e_i, t) \in \mathcal{F}_t\}$, $\mathcal{N}_{i,t}^{\text{tri}} = \{(e_j, r_j) \mid (e_j, r_j, e_i, t) \in \mathcal{F}_t\}$, and φ is used for generating the combined information of entity-relation pairs. In our model, φ_{ij} is obtained by element-wise multiplication of entity vector $\boldsymbol{h}_{i,t}$ and relation vector $\boldsymbol{r}_{j,t}$. The SE is obtained by merging the above three levels of SE into the original embedding. Thus, the output of the layer is

$$\boldsymbol{h}'_{i,t} = \boldsymbol{h}_{i,t} + \boldsymbol{h}_{i,t}^{\text{rel}} + \boldsymbol{h}_{i,t}^{\text{ent}} + \boldsymbol{h}_{i,t}^{\text{tri}}. \tag{4}$$

For the embedding vector $\boldsymbol{r}_{i,t}$ of relation r_i, the semantics at the current timestamp are affected by the semantics at the previous timestamp; therefore, unlike SEGNN, in which the embeddings of the relations in each layer are randomly initialized, we use the relation embedding of the previous timestamp and combine it using mean pooling from the entity embedding as the input embedding of the current relation: $\boldsymbol{r}'_{i,t} = \boldsymbol{W}_r[\text{pooling}\left(\boldsymbol{H}_{t-1}, \mathcal{N}_{i,t}^{\text{ent}}\right); \boldsymbol{r}_{i,t}]$, $\boldsymbol{W}_r$ is a linear matrix.

Recurrent unit. After obtaining the information of the SE at a certain timestamp, to go deeper into the semantic information at different timestamps, we input the embedding obtained from the SE layer into the recurrent unit. Considering that stacking multiple layers of graph neural network (GNN) may lead to over-smoothing [41] problem, we use a residual layer to solve this problem:

$$\hat{\boldsymbol{H}}_t = \boldsymbol{U}_t^e * \boldsymbol{H}'_t + (1 - \boldsymbol{U}_t^e) * \boldsymbol{H}_{t-1}, \boldsymbol{U}_t^e = \sigma\left(\boldsymbol{H}_{t-1}\boldsymbol{W}_e + \boldsymbol{b}_e\right), \tag{5}$$

where $\boldsymbol{U}_t^e \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$, $\boldsymbol{W}_e \in \mathbb{R}^{d\times|\mathcal{E}|}$ is a linear matrix, and "*" denotes the dot product operation. Then we feed the entity matrix into the GRU:

$$\boldsymbol{H}_t = \text{GRU}(\hat{\boldsymbol{H}}_t, \boldsymbol{H}'_t). \tag{6}$$

Similarly, there are symmetric operations for relations to obtain $\boldsymbol{R}_t$.

3.3 Historical semantic evidence encoder

When considering history, we first introduce the concept of three levels of the history of the fact: For a certain fact, $u = (s_u, r_u, o_u, t_u)$, the relation-level history of u refers to $\mathcal{H}_u^{\text{rel}} = \{(s_u, r_u, t) | (s_u, r_u, *, t) \in \mathcal{F}, t < t_u\}$, u's entity-level history means $\mathcal{H}_u^{\text{ent}} = \{(s_u, o_u, t) | (s_u, *, o_u, t) \in \mathcal{F}, t < t_u\}$, and u's triple-level history refers to $\mathcal{H}_u^{\text{tri}} = \{(s_u, r_u, o_u, t) | (s_u, r_u, o_u, t) \in \mathcal{F}, t < t_u\}$. "$*$" refers to any entity for $\mathcal{H}_u^{\text{rel}}$ or any relation for $\mathcal{H}_u^{\text{rel}}$.

We believe that when considering history for a prediction task, to measure the relative importance of recent facts and distant facts, not only the frequency of history occurrence but also the time difference should be considered. Namely, for a fact with timestamp t, if a certain fact appears many times before t, or if some fact occurs at a timestamp closer to t, there will be a higher probability that the corresponding event will occur at t. To this end, we introduce the Hawkes process [6], which assumes that events that occurred in the past have a positive impact on the current event. The more occurrences and the closer those occurrences, the greater the positive impact, and the impact decays with time. The Hawkes process intensity function is

$$p(u|\mathcal{H}_u^*) = \lambda_u^*(t) = \mu + \sum_{v:t_v<t} g\left(t - t_v\right), \tag{7}$$

where $\mathcal{H}_u^*$ can be $\mathcal{H}_u^{\text{rel}}$, $\mathcal{H}_u^{\text{ent}}$, or $\mathcal{H}_u^{\text{tri}}$, $\lambda_u^*(t)$ corresponds to $\lambda_u^{\text{rel}}(t)$, $\lambda_u^{\text{ent}}(t)$, or $\lambda_u^{\text{tri}}(t)$, which refers to the probability of fact u occurring at time t given the known relation-level, entity-level, or triple-level history

of u, and we call them historical relation-level SE, historical entity-level SE, and historical triple-level SE. The three levels of HSE are collectively referred to as HSE, μ is the background intensity, $v \in \mathcal{H}_u^*$ represents the history, t_v represents the timestamp of v, and $g(\cdot)$ is a triggering kernel function. Our model sets $g(\cdot)$ as a commonly used exponential function. Since the SE between different facts has been considered in the TSE encoder, we only consider the repetition of the facts, so the HSE can be modified as follows:

$$\lambda_u^*(t) = \begin{cases} 1 + \sum_{v \in \mathcal{H}_u^*} \mathrm{e}^{-(t-t_v)/d}, & \mathcal{H}_u^* \neq \phi, \\ 0, & \mathcal{H}_u^* = \phi, \end{cases} \tag{8}$$

where $d = T/K$ is the time span. Because the time granularity of each dataset is different, we divide the dataset into K equal parts according to the timestamp, and the facts within each equal part are regarded as happening at the same time. For a fact that occurred at a certain timestamp, the larger K is, the smaller d is, so the weight $\mathrm{e}^{-(t-t_v)/d}$ of this fact is larger, which means the fact is more important. The Hawkes process assigns attenuation weights to facts based on their time of occurrence within each time span, with more recent facts receiving higher weights than those occurring further in the past. Finally, we calculate the cumulative impact weight of a past fact on the current fact. By this way, we can investigate the relative importance of facts by adjusting the value of hyperparameter K. We will give more analysis of K in the experimental part.

For an entity prediction $q = (s_q, r_q, ?, t_q)$, we first fill in all possible entities to obtain a candidate set of all possible entity prediction answers $A_q = \{u_q = (s_q, r_q, o, t_q) | o \in \mathcal{E}\}$. We calculate the HSE of each quadruple u_q in set A_q, which is written as $\lambda_q^* \in \mathbb{R}^{|\mathcal{E}|}$. Since there are $|\mathcal{E}|$ subject entities and $|\mathcal{R}|$ object entities, the entity prediction at time t has $|\mathcal{E}| \times |\mathcal{R}|$ possible queries, so the entity prediction's HSE matrix of all possible queries at timestamp t can be obtained by stacking λ_q^* for $|\mathcal{E}| \times |\mathcal{R}|$ possible q, which is written as $\boldsymbol{\Lambda}_t^{*,\mathrm{ep}} \in \mathbb{R}^{(|\mathcal{E}|\times|\mathcal{R}|)\times|\mathcal{E}|}$. For example, the $(i \times j, k)$ element of $\boldsymbol{\Lambda}_t^{\mathrm{tri,ep}}$ represents historical triple-level SE of $u = (e_i, r_j, e_k, t)$ at time t. Because the historical triple-level SE can better represent the exact events that have recurred in history, we only consider the historical entity-level SE when the historical triple-level SE is 0 for a certain possible query.

Furthermore, to reduce complexity, we only reserve the top-n historical entity-level SE for each query. To combine the two types of SE, we normalize the historical entity-level SE to the historical triple-level SE by $\max(\boldsymbol{\Lambda}_t^{\mathrm{tri,ep}})/\max(\boldsymbol{\Lambda}_t^{\mathrm{ent,ep}})$. Then we can get HSE of all possible entity predictions at time t as

$$\boldsymbol{\Lambda}_t^{\mathrm{ep}} = \boldsymbol{\Lambda}_t^{\mathrm{tri,ep}} + \frac{\max(\boldsymbol{\Lambda}_t^{\mathrm{tri,ep}})}{\max(\boldsymbol{\Lambda}_t^{\mathrm{ent,ep}})} \boldsymbol{\Lambda}_t^{\mathrm{ent,ep}} * \boldsymbol{M}^{\mathrm{ep}}, \boldsymbol{M}_{ij}^{\mathrm{ep}} = \begin{cases} \mathbf{1}, & \boldsymbol{\Lambda}_i^{\mathrm{tri,ep}} = \mathbf{0}, \\ \mathbf{0}, & \boldsymbol{\Lambda}_i^{\mathrm{tri,ep}} \neq \mathbf{0}, \end{cases} \tag{9}$$

where $\boldsymbol{M}^{\mathrm{ep}} \in \mathbb{R}^{(|\mathcal{E}|\times|\mathcal{R}|)\times|\mathcal{E}|}$ represents the mask, and $\boldsymbol{M}_{ij}^{\mathrm{ep}}, \boldsymbol{\Lambda}_{ij}^{\mathrm{tri,ep}}$ represent the i-th row and the j-th column of $\boldsymbol{M}^{\mathrm{tri,ep}}$ and $\boldsymbol{\Lambda}^{\mathrm{tri,ep}}$. Similarly, we can obtain the relation prediction HSE matrix $\boldsymbol{\Lambda}^{\mathrm{rp}} \in \mathbb{R}^{(|\mathcal{E}|\times|\mathcal{E}|)\times|\mathcal{R}|}$, and the $(i \times j, k)$ element of $\boldsymbol{\Lambda}_t^{\mathrm{rp}}$ represents the HSE of $u = (e_i, r_k, e_j, t)$ for relation prediction.

3.4 SE-aware decoder

For the decoder of our model, we use ConvTransE, which has been proven effective for TKG reasoning by the RE-GCN and TiRGN for TKG reasoning. To use the evolutional SE obtained by the TSE encoder and the HSE obtained by the HSE encoder, we use Conv-TransE to decode the information obtained by the two encoders. Specifically, we employ two separate Conv-TransE models to capture the information within TSE and HSE, respectively. In regards to TSE, we obtain the temporal probability vector for entity prediction $(s, r, ?, t+1)$ and from the TSE encoder:

$$\boldsymbol{p}_{\mathrm{temp}}^{\mathrm{ep}} = \boldsymbol{p}_{\mathrm{temp}}\left(o \mid s, r, \boldsymbol{H}_t, \boldsymbol{R}_t\right) = \mathrm{softmax}\left(\boldsymbol{H}_t \, \mathrm{ConvTransE}\left(\boldsymbol{h}_t, \boldsymbol{r}_t\right)\right). \tag{10}$$

We can also get the relation prediction $(s, ?, o, t+1)$, which is $\boldsymbol{p}_{\mathrm{temp}}^{\mathrm{rp}} = \mathrm{softmax}\left(\boldsymbol{R}_t \, \mathrm{ConvTransE}\left(\boldsymbol{h}_t, \mathbf{o}_t\right)\right)$.

In the same way, we can obtain the historical probability vector for the entity prediction from the HSE encoder:

$$\boldsymbol{p}_{\mathrm{his}}^{\mathrm{ep}} = \boldsymbol{p}_{\mathrm{his}}\left(o \mid s, r, \boldsymbol{H}_t, \boldsymbol{R}_t\right) = \mathrm{softmax}\left(\boldsymbol{H}_t \, \mathrm{ConvTransE}\left(\boldsymbol{h}_t, \boldsymbol{r}_t\right) * \boldsymbol{\Lambda}_{sr}^{\mathrm{ep}}\right). \tag{11}$$

Table 1 Details of the six widely used TKG datasets. $\mathcal{R}'$, $\mathcal{E}'$ represent the unseen relation set and the unseen entity set in the test set. The date format is dd/mm/yyyy or yyyy.

	ICEWS14	ICEWS18	ICEWS0515	WIKI	YAGO	GDELT
Time interval	1 d	1 d	1 d	1 y	1 y	15 min
Duration	364 d	303 d	11 y	231 y	188 y	30 d
Start time	01/01/2014	01/01/2018	01/01/2005	1786	1830	01/01/2018
End time	12/31/2014	10/31/2018	12/31/2015	2017	2017	01/31/2018
$\lvert\mathcal{E}\rvert$	7128	23033	10488	12435	10585	7691
$\lvert\mathcal{R}\rvert$	230	256	251	24	10	240
$\lvert\mathcal{F}\rvert$	90730	468558	461329	669934	201089	2278405
$\lvert\mathcal{E}'\rvert/\lvert\mathcal{E}\rvert$	5.86%	4.51%	3.66%	6.00%	3.35%	3.80%
$\lvert\mathcal{R}'\rvert/\lvert\mathcal{R}\rvert$	1.30%	0.78%	0.40%	0.00%	0.00%	0.40%

Similarly, we can get the historical probability vector of the relation prediction, and the result is calculated as $\boldsymbol{p}_{\text{his}}^{\text{rp}} = \text{softmax}\left(\boldsymbol{R}_t \,\text{ConvTransE}\left(\boldsymbol{h}_t, \boldsymbol{o}_t\right) * \boldsymbol{\Lambda}_{so}^{\text{rp}}\right)$.

To combine TSE and HSE, we use the weighted sum of the TSE and HSE, and use $\eta \in [0,1]$ to represent the weight of the HSE, so we get the final score function of entity prediction:

$$\boldsymbol{p}_{\text{final}}^{\text{ep}} = \eta \boldsymbol{p}_{\text{his}}^{\text{ep}} + (1-\eta)\boldsymbol{p}_{\text{temp}}^{\text{ep}}. \tag{12}$$

Symmetrically, we can obtain the final scoring function for relation prediction by $\boldsymbol{p}_{\text{final}}^{\text{rp}} = \eta \boldsymbol{p}_{\text{his}}^{\text{rp}} + (1-\eta)\boldsymbol{p}_{\text{temp}}^{\text{rp}}$.

3.5 Parameter learning

Entity and relation prediction can be regarded as multi-label classification problems, that is, for known facts from time 0 to T, we need to classify entities for query $(s, r, ?, T+1)$ or relations for query $(s, ?, o, T+1)$ at $t+1$. Therefore, for $t \in [0, T-1]$, we learn the HSE and the evolutional patterns of $[t-m+1, t]$. At the time $t+1$, we perform a multi-label classification task to learn the parameters, and we use γ to represent the proportion between the entity prediction and relation prediction, $\gamma \in [0,1]$. The loss functions of the TSE can be expressed as follows:

$$\begin{aligned} L_{\text{temp}} &= \gamma L_{\text{temp}}^{e} + (1-\gamma) L_{\text{temp}}^{r} \\ &= \sum_{t=0}^{T-1} \sum_{q \in \mathcal{F}_{t+1}} \left(\gamma \boldsymbol{y}_{t+1}^{e} \log \boldsymbol{p}_{\text{temp}}^{\text{ep}} + (1-\gamma)\boldsymbol{y}_{t+1}^{r} \log \boldsymbol{p}_{\text{temp}}^{\text{rp}}\right), \end{aligned} \tag{13}$$

where $q = (s, r, o, t+1)$, $\boldsymbol{y}_{t+1}^{r} \in \mathbb{R}^{|\mathcal{E}|}, \boldsymbol{y}_{t+1}^{e} \in \mathbb{R}^{|\mathcal{E}|}$ are label vectors for the relation and entity prediction. If a fact occurs at $t+1$, the corresponding element that represents the answer entity or relation is set to 1; otherwise, it is set to 0. We can also get the loss function of HSE by $L_{\text{hist}} = \gamma L_{\text{hist}}^{e} + (1-\gamma) L_{\text{hist}}^{r}$. Finally, we obtain the total loss $\mathcal{L}$ using the weighted sum of the TSE's loss and HSE's loss:

$$\mathcal{L} = \eta L_{\text{hist}} + (1-\eta) L_{\text{temp}}. \tag{14}$$

4 Experiments

In this section, we begin by introducing the experimental setup. Subsequently, we present the experimental results and conduct ablation studies. Following that, we delve into a thorough analysis of our model, which includes exploring the replacement of internal methods within our model and assessing the impact of integrating our module into other baseline models. Then, we observe the training overhead of our model in terms of computational complexity and actual training time. Through case studies, we confirm the effectiveness of TSE and HSE in addressing the challenge of unseen entities or relations. Finally, we showcase our model's performance in predicting unseen entities.

Table 2 Optimal hyperparameter settings of the six datasets.

	ICEWS14	ICEWS18	ICEWS0515	WIKI	YAGO	GDELT
K	20	40	20	80	60	20
m	8	10	15	2	1	6
η	0.3	0.3	0.3	0.5	0.5	0.5
γ	0.7	0.6	0.7	0.6	0.6	0.6

4.1 Experiment setup

4.1.1 *Datasets*

Previous work mainly used four datasets, namely the GDELT [42], WIKIDATA [43], YAGO [44], and ICEWS [45] datasets. Generally, previous work used three subsets of the ICEWS dataset: ICEWS05-15 [24], ICEWS18 [1], ICEWS14 [24], and a subset of the other three datasets following [1]. These datasets can be divided into two groups. (1) Event-based TKGs, which include the ICEWS and GDELT datasets, meaning that the facts are mostly instant actions. (2) Public TKGs, which include the YAGO and the WIKI datasets, have meta-facts like $(s, r, o, [t_s, t_e])$, which occur within a period between t_s and t_e, where t_s, t_e are the start timestamp and end timestamp of the fact, respectively. This makes the datasets have many of the same facts at adjacent timestamps $[(s, r, o, t_s), (s, r, o, t_s + 1), \ldots, (s, r, o, t_e)]$. In previous work, each dataset has been divided into a training set, a validation set, and a test set by timestamp in time order with proportions of 80%, 10%, and 10%, respectively, following [1]. We also use the same dataset settings. More detailed information on these datasets is illustrated in Table 1.

4.1.2 *Baselines*

For the entity prediction task, we compare across three types of models: static graph reasoning (convE [12], DistMult [10], ComplEx [11]), Interpolation TKG reasoning (TTransE [22], TA-DistMult [24], DE-SimplE [46], TNTComplEx [25]), and our focus, extrapolation TKG reasoning (RE-NET [1], CyGNet [4], TANGO [47], xERTE [29], RE-GCN [2], TiTer [27], TiRGN [31]).

4.1.3 *Evaluation metrics*

We use the mean reciprocal rank (MRR) and hit@k which are widely used in evaluating TKG reasoning models. The MRR is the average of the reciprocal ranks of all true candidate entities/relations for the query and the hit@k represents the proportion of true entities/relations in the top-k predicting candidates. There exist two types of filtering settings to evaluate the MRR and hit@k: static filtered metrics [7] and time-aware filtered metrics [29]. Some studies have proposed that it is not appropriate to use static filtering settings [48] for TKG reasoning, and it is more reasonable to use time-aware filtered metrics to evaluate extrapolate reasoning in the TKG [29], so we use the time-filtered MRR and hit@k to evaluate our model.

4.1.4 *Hyperparameters*

In our model, the dimension of entities' embedding and relations' embedding d is set to 200. The number of ConvTransE kernels is set to 50, and the kernel size is set to 2×3. Considering that RE-GCN [2] and TiRGN [31] are the most related models, both of which incorporate static graph constraints into all three ICEWS datasets, for a fair comparison, we also adopt these constraints. We pick the top 3 entity/relation-level HSE in the HSE encoder. The number of the TSE encoder's recurrent units m, the number of timestamp segments K, the HSE loss's weight η, and the entity prediction loss's weight γ for the six datasets are shown in Table 2. Finally, we use the Adam optimizer to update the parameters, and the learning rate is set to 0.001. All experiments are run in Tesla V100 or RTX 3090 with Pytorch [49].

4.2 Experimental results

Through experiments we can see that the performance of static graph reasoning models and TKG reasoning models with the interpolation setting are far lower than the methods for reasoning TKG with the extrapolation setting, so we focus on TKG reasoning SOTA models with extrapolation setting for comparison. The entity prediction results are presented in Table 3 and the relation prediction results are presented in Table 4.

Table 3 Entity prediction results on the six datasets (%). The best results are in bold and the second best results are underlined.

Model	ICEWS14				ICEWS18				ICEWS0515			
	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10
ConvE	34.50	24.83	38.56	53.88	24.51	16.23	29.25	44.51	33.81	24.78	39.00	54.95
Distmult	27.67	18.16	31.15	46.96	10.17	4.52	10.33	21.25	28.73	19.33	32.19	47.54
ComplEx	30.84	21.51	34.48	49.59	21.01	11.87	23.47	39.97	31.69	21.44	35.74	52.04
TTransE	13.43	3.11	17.32	34.55	8.31	1.92	8.56	21.89	15.71	5.00	19.72	38.02
TA-DistMult	26.47	17.09	30.22	45.41	16.75	8.61	18.41	33.59	24.31	14.58	27.92	44.21
DE-SimplE	32.67	24.43	35.69	49.11	19.30	11.53	21.86	34.80	35.02	25.91	38.99	52.75
TNTComplEx	32.12	23.35	36.03	49.13	21.23	13.28	24.02	36.91	27.54	19.52	30.80	42.86
RE-NET	39.86	30.11	44.02	58.21	29.78	19.73	32.55	48.46	43.67	33.55	48.83	62.72
CyGNet	37.65	27.43	42.63	57.90	27.12	17.21	30.97	46.85	40.42	29.44	46.06	61.60
TANGO	–	–	–	–	28.97	19.51	32.61	47.51	42.86	32.72	48.14	62.34
xERTE	40.79	32.70	45.67	57.30	29.31	21.03	33.51	46.48	46.62	37.84	52.31	63.92
RE-GCN	42.00	31.63	47.20	61.65	32.62	22.39	36.79	52.68	48.03	37.33	53.90	68.51
TiTer	41.73	32.74	46.46	58.44	29.98	22.05	33.46	44.83	47.60	38.29	52.74	64.86
TECHS	43.88	34.59	49.36	61.95	30.85	21.81	35.39	49.82	48.38	38.34	54.69	68.92
TiRGN	44.04	33.83	48.95	63.84	33.66	23.19	37.99	54.22	50.04	39.25	56.13	70.71
RPC	44.55	34.87	49.80	65.08	34.91	24.34	38.74	54.22	51.14	39.47	57.11	bf 1.75
Ours	**45.81**	**35.79**	**51.01**	**64.66**	**34.94**	**24.57**	**39.44**	**55.08**	**51.54**	**41.05**	**57.62**	71.19

Model	WIKI				YAGO				GDELT			
	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10
ConvE	14.52	11.44	16.36	22.36	–	–	–	–	16.55	11.02	18.88	31.60
Distmult	10.89	8.92	10.97	16.82	11.98	10.20	12.31	14.93	5.50	0.47	4.94	15.25
ComplEx	24.47	19.69	27.28	34.83	9.84	5.17	9.58	18.23	16.96	11.25	19.52	32.35
TTransE	29.27	21.67	34.43	42.39	5.68	1.42	9.04	11.21	15.71	5.00	19.72	38.02
TA-DistMult	44.53	39.92	48.73	51.71	11.50	10.21	11.90	13.88	12.00	5.76	12.94	23.54
DE-SimplE	45.43	42.60	47.71	49.55	11.73	10.70	12.10	13.51	19.70	12.22	21.39	33.70
TNTComplEx	45.03	40.04	49.31	52.03	12.00	11.12	12.13	13.57	19.53	12.41	20.75	33.42
RE-NET	58.32	50.01	61.23	73.57	66.93	58.59	71.48	86.84	19.55	12.38	20.80	34.00
CyGNet	58.78	47.89	66.44	78.70	68.98	58.97	76.80	86.98	20.22	12.35	21.66	35.82
TANGO	53.04	51.52	53.84	55.46	63.34	60.04	65.19	68.79	19.66	12.50	20.93	33.55
xERTE	73.60	69.05	78.03	79.73	84.19	80.09	88.02	89.78	19.45	11.92	20.84	34.18
RE-GCN	78.53	74.50	81.59	84.70	82.30	78.83	84.27	88.58	19.69	12.46	20.93	33.81
TiTer	73.91	71.70	75.41	76.96	87.47	80.09	89.96	90.27	18.19	11.52	19.20	31.00
TECHS	75.98	–	–	82.39	89.24	–	–	92.39	–	–	–	–
TiRGN	81.65	77.77	85.12	87.08	87.95	84.34	91.37	92.92	21.67	13.63	23.27	37.60
RPC	81.18	76.28	85.43	88.71	88.87	85.10	92.57	**94.04**	22.41	14.42	24.26	38.33
Ours	**81.94**	**77.85**	**85.78**	**87.28**	**90.55**	**88.34**	**92.67**	93.09	**24.79**	**16.18**	**27.34**	**41.54**

Table 4 Relation prediction results on the six datasets (%). The best results are in bold and the second best results are underlined.

Model	ICE14	ICE18	ICE05-15	WIKI	YAGO	GDELT
RGCRN	41.92	42.19	44.18	96.47	95.30	20.99
RE-GCN	45.29	45.66	47.07	98.90	98.80	21.47
TiRGN	47.71	46.64	48.17	99.04	**99.30**	24.91
RE-SEGNN	**48.63**	**48.09**	**49.95**	**99.07**	98.57	**28.53**

The reasons that our model outperforms CyGNet [4] and RE-NET [1] are as follows: CyGNet considers the frequency of history to learn long-term information but ignores the time order of historical facts, which affects the trend of what fact will occur. RE-NET only considers historical facts within a period using the RNN and obtains structural dependencies through mean pooling, which may lead to the problem of vanishing or exploding gradients. The problems that RE-NET and CyGNet face can be easily solved by our model's TSE encoder and HSE encoder, respectively. TiTer [27] and xERTE [29] can achieve good performance when the total number of timestamps is small, such as in the WIKI and YAGO datasets, but when facing timestamps on a large scale, it is difficult to make TiTer's reinforcement learning agent reach the correct answer. TiRGN [31], assumed that it is not necessary to consider the frequency, and needs

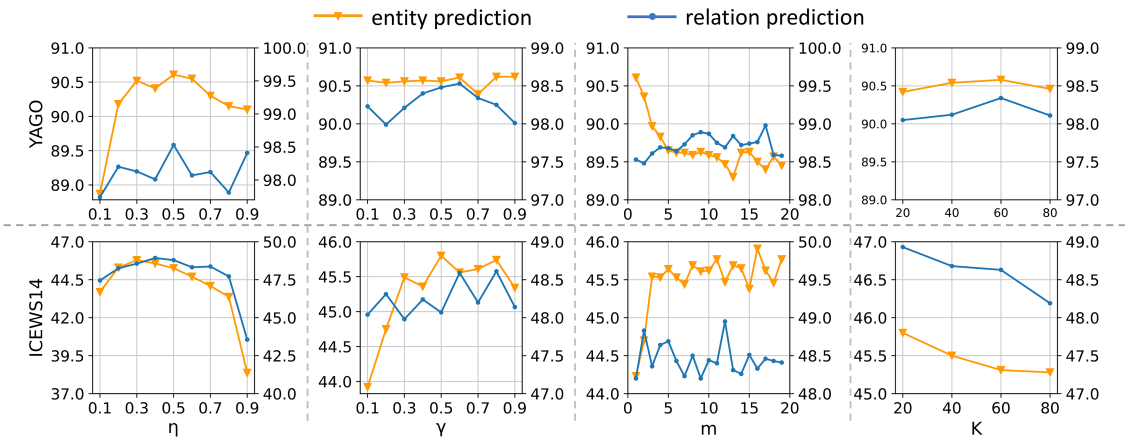

Figure 3 (Color online) Sensitivity analysis of η, γ, m, and K. In each subgraph, the left y-axis is the MRR (%) for entity prediction, and the right y-axis is the MRR (%) for relation prediction.

Table 5 RE-SEGNN ablation studies.

Model	ICEWS14				YAGO			
	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10
TSE	40.08	30.44	44.41	58.67	81.59	77.70	84.75	88.06
HSE	42.19	32.68	46.66	60.28	81.23	77.35	84.28	87.91
RNN+SEGNN	36.61	27.06	40.55	54.94	50.75	47.02	52.45	57.23
Ours (TSE+HSE)	45.80	35.92	50.70	64.39	90.61	88.47	92.73	93.08

to know whether a fact has occurred in history. On the contrary, by considering the facts' frequency in history and time order, our model can perform better on the entity and relation prediction task.

However, our model cannot obtain better results for relation prediction on the YAGO dataset and has only a minor improvement compared to the SOTA models on the WIKI dataset because of the following reasons. (1) Table 1 shows that there are no unseen relations in the WIKI and YAGO test sets, and our model focuses more on unseen relations. (2) The WIKI and YAGO datasets are public KGs, making repetitive historical facts far more important than time order and frequency. In contrast, the ICEWS and GDELT datasets are event-based datasets, and both of them do not have many meta-facts, so our model can obtain better predictions than the SOTA models. (3) As shown in Table 1, the number of relations in the WIKI and YAGO datasets is too small, which makes the number of entities connected to each relation very large, making it more difficult for our TSE encoder to obtain useful structural dependencies from entities compared to the TiRGN and RE-GCN.

4.3 Sensitivity analysis

We perform a sensitivity analysis on γ, which represents the proportion of entity prediction loss, and η, which represents the proportion of HSE. We also perform a sensitivity analysis on the number of time segments K and the history length m to show the performance change. The results are shown in Figure 3, where the first and the second rows show the results on the YAGO dataset and the ICEWS14 dataset, respectively.

From the sensitivity analysis of η, we can see that the TSE encoder and HSE encoder complement each other. The sensitivity analysis of γ shows that considering both entity and relation prediction can improve the overall performance. The sensitivity analysis of m shows that for the public KG datasets, facts that occurred more recently have a greater effect because there exist many meta-facts that can continue for a period of time, and after that period, the same fact rarely happens again.

The results in Figure 3 show that the parameter K plays a key role in representing changes in the impact of historical facts over time. We can see that YAGO is a public database characterized by events spanning a period of time. When we make predictions, the facts at the recent timestamps may not have ended yet, so recent facts are more valuable for prediction. A larger K allows us to capture the impact of recent facts on YAGO. As for ICEWS14, it consists of event-based knowledge that occurs instantaneously. In that case, the recent facts may not contribute significantly to predictions, so a smaller K allows us to

Table 6 Effectiveness of TSE and HSE.

Model	ICEWS14				YAGO			
	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10
TSE.r.gcn	36.48	27.06	40.02	55.52	50.40	46.64	52.05	56.88
TSE	40.08	30.44	44.41	58.67	81.59	77.70	84.75	88.06
TSE.r.gcn+HSE	42.30	32.94	46.74	60.12	81.36	77.45	84.39	88.04
HSE	42.19	32.68	46.66	60.28	81.23	77.35	84.28	87.91
TSE+HSE.r.once	39.10	29.36	43.25	58.74	59.20	50.32	63.77	77.94
TSE+HSE.r.freq	39.68	30.24	44.13	57.54	75.10	70.59	78.25	82.99
SEGNN	30.22	21.04	34.34	48.01	38.56	35.01	40.34	45.24
TiRGN+HSE	44.52	34.33	49.45	64.24	88.15	85.54	92.61	93.03
TiRGN.r.segnn	43.13	32.24	48.21	62.34	86.81	82.93	89.82	91.94
RE-GCN+HSE	42.69	32.03	48.78	62.74	83.12	80.45	86.23	90.87
RE-GCN.r.segnn	40.79	30.35	46.58	60.78	81.47	78.02	84.54	88.12
Ours (TSE+HSE)	45.80	35.92	50.70	64.39	90.61	88.47	92.73	93.08

obtain more semantic information of historical facts. This observation further emphasizes the adaptability of our model in balancing the consideration of distant and recent facts.

4.4 Ablation studies

We perform ablation studies to explore the effectiveness of each part of our model on the ICEWS14 and YAGO datasets. The results are shown in Table 5, in which HSE indicates the HSE encoder, TSE indicates the TSE encoder, RNN+SEGNN indicates simply integrating the RNN and SEGNN models to make the prediction.

From the results, we can observe that RNN+SEGNN performs poorly due to the lack of consideration for temporal information. Both HSE and TSE show significant improvements over RNN+SEGNN when used individually. The combination of HSE and TSE is even more effective as they complement each other. Moreover, we can see that since SEGNN is a static graph model, it cannot capture temporal information in temporal KG as TSE does, so the performance of RNN+SEGNN in prediction is not very good.

4.5 Module analysis

To further analyze the effectiveness of our module, we conduct three parts of experiments. The results are shown in Table 6.

In the first part, we replaced the time-aware SEGNN in TSE with RGCN, and replaced the Hawkes process in HSE with other statistical methods. TSE.r.gcn means we replace the time-aware SE layer in the TSE encoder with a one-layer GCN, HSE.r.once means HSE encoder only considers whether the semantic evidence happened similar to the TiRGN [31], and HSE.r.freq means HSE encoder only considers the frequency of HSE.

From the results, whether the SE layer in a separate TSE is replaced with gcn (TSE.r.gcn) or the SE layer in RE-SEGNN is replaced with gcn (TSE.rgcn+HSE), the effect will significantly decrease, indicating the effectiveness of TSE. In terms of HSE, replacing the statistical methods in HSE with only counting whether a certain event has occurred (TSE+HSE.r.once) similar to TiRGN, or replacing the statistical methods in HSE with frequency counting methods similar to CyGnet (TSE+HSE.r.freq), both do not perform well in RE-SEGNN using Hawkes processes.

In the second part, to demonstrate the superiority of our proposed TSE encoder in acquiring semantic evidence from TKG compared to SEGNN, we conducted experiments using SEGNN. The results show that the performance of SEGNN is significantly lower than TSE. This observation emphasizes that while SEGNN is proficient at extracting semantic evidence in KGs, it faces challenges when applied to TKG. SEGNN captures static semantic evidence on the entire dataset and cannot adapt to the dynamic characteristics of TKG, while TSE captures the semantic evidence at each timestamp, and utilizes GRU to model the semantic evolution process. Therefore, TSE is more suitable for semantic evidence extraction in TKG.

Table 7 Complexity analysis of RESEGNN and SOTA models.

Model	Complexity
RE-GCN	$O(mn(\|\mathcal{D}\|\omega + \|\mathcal{R}\|R_e))$
TiRGN	$O(mn(\|\mathcal{D}\|\omega + \|\mathcal{R}\|R_e) + \|\mathcal{F}\|)$
RE-NET	$O(mn(\|\mathcal{D}\|\omega + \|\mathcal{R}\|\|\mathcal{E}\|))$
RE-SEGNN	$O(mn(\|\mathcal{D}\| + \|\mathcal{R}\|R_e)) + O(\|\mathcal{F}\|\|\mathcal{T}\|)$

Table 8 Training time and epochs of some SOTA models on ICEWS14.

Model	Training time (each epoch)	Training epochs	Best epochs	Training time
RE-GCN	35±5 s	100±50	12±5	≈ 50 min
TiRGN	4±0.5 min	50±5	15±5	≈ 60 min
RE-NET	12±3 min	50±10	20±5	≈ 180 min
Ours	6±0.5 min (TSE), 30±10 min (HSE)	10±5	10±5	≈ 90 min (HSE+TSE)

In the third part, to analysis the impact of our modules to other models, we conducted experiments by integrating them with other RNN-based models. The terms 'TiRGN+HSE' and 'RE-GCN+HSE' refer to the integration of the respective model with HSE. Additionally, '.r.segnn' signifies the substitution of the graph neural network in the corresponding model with segnn thereby emulating the behavior of TSE. The results of these experiments reveal that our HSE module contributes to a noticeable improvement in the SOTA RNN-based model, whereas TSE demonstrates relatively modest enhancements in these models. This observation is primarily attributed to the inherent limitations of these models in effectively modeling and processing semantics.

4.6 Complexity and training time analysis

4.6.1 *Computational complexity*

To analyze the efficiency of our proposed RE-SEGNN, we analyzed the computational complexity of our training model, as shown in Table 7. For TSE, the time complexity of each time t is $O(|\mathcal{D}|)$, where $|\mathcal{D}|$ refers to the maximum number of concurrent facts in all historical timestamps, that is, $|\mathcal{D}| = \arg\max(|\mathcal{F}_t|), t \in \mathcal{T}_{\text{train}}$. The computational complexity of the pooling and residual layers at each timestamp is $O(|\mathcal{R}|R_e)$, where $|\mathcal{R}|$ represents the size of the relation set, and R_e represents the maximum number of entities associated with a single relation at all timestamps. Since we have m time units, if we perform n epochs training, the total complexity of TSE is $O(mn(|\mathcal{D}| + |\mathcal{R}|R_e))$. As for HSE, for each fact, our model needs to update the corresponding weight of $\mathbf{\Lambda}^{\text{ep}}$ and $\mathbf{\Lambda}^{\text{rp}}$, so the complexity is $O(|\mathcal{F}|)$. But HSE does not need to train, it only needs to perform one preprocessing. If we perform n epochs of training, our overall time complexity is $O(mn(|\mathcal{D}| + |\mathcal{R}|R_e) + |\mathcal{F}||\mathcal{T}|)$. It is worth noting that when the number of training epochs n and the historical unit m considered become larger, $|\mathcal{F}||\mathcal{T}|$ does not account for a large proportion. Meanwhile, we calculate the computational complexity of some SOTA models. We can find that although our model has greater complexity than other models, the complexity is tolerable.

4.6.2 *Training time*

To prove that the time our model spends is acceptable, we also compare the training time with other SOTA models on ICEWS14. In Table 8, we show the training time of several SOTA models, and all the models are run in the same environment with best hyperparameters. The second column shows the time spent on each epoch of training of the model, the third column shows the number of training epochs that reach the termination condition, the fourth column shows the number of epochs that achieve the best result on the validation set, and the fifth column shows the training time.

For the baseline models, the total number of training epochs is quite different from the number of epochs needed to achieve the best results on the validation set, indicating an obvious waste of training time. In contrast, our model achieves the best results on the training set and validation set at similar epochs, indicating that although the per-epoch training time of our model is relatively long, the number of training epochs can be significantly reduced; consequently, the final total training time is acceptable.

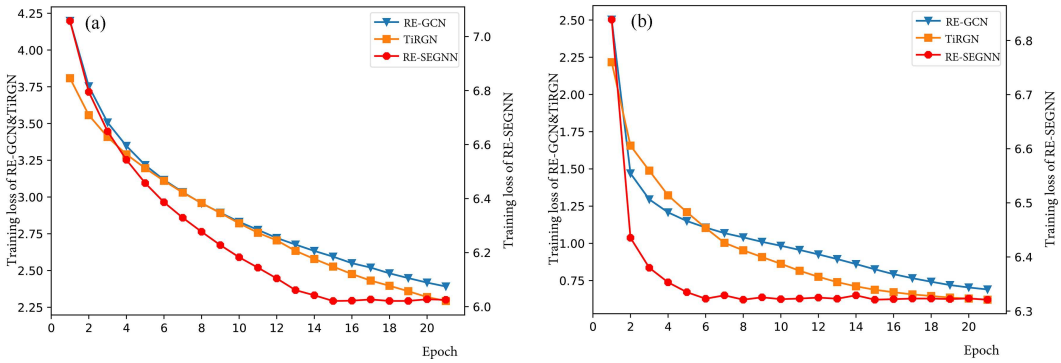

Figure 4 (Color online) Loss curves of some SOTA models. (a) ICEWS14; (b) YAGO.

4.6.3 *Loss curves*

We draw the training loss curves of several models using recurrent units on ICEWS14 and YAGO in Figure 4. We can see that our model's loss curve drops the fastest and is the first to stabilize. The loss curve of other models takes a longer time to reach stability and reaches stability. The reason why our training can reach stability quickly is that our model uses not only TSE (RNN-based model), but also the HSE based on the Hawkes process. Although HSE will take up a certain amount of computing overhead, it can effectively improve the accuracy of our model and prevent overfitting.

4.7 Case studies

To better demonstrate our model's modeling of unseen entities and relations, we extract several entity prediction cases from the ICEWS14 dataset which contains unseen entities or relations, as shown in Table 9.

We divided the case study into three distinct parts, analyzing entity-level SE, relation-level SE, and triple-level SE, individually. In Table 9, the 'query' row represents the question we intend to reason about, and the correct answer is indicated to the right of the arrow. The 'history' row displays recent related facts that occurred just before the query time. Taking the entity level SE as an example, the entity "Boko Haram" has never occurred in history, which we call the unseen entity, and the relation "Abduct, hijack, or take hostage" most recent occurrence in history is displayed in the history row. Our model obtains the probability of predicting entities through TSE and HSE, respectively. We display the candidates with the highest probability in the candidate row.

We can find that the candidates who get the best scores are not always the most frequently occurred events, which indicates that we can better model the changing event history considering not just by frequency but also the time difference between the history events and the current time. Furthermore, when facing unseen entities, our model can still get the scores to make the right prediction by fine-grained levels of SE.

4.8 Prediction of unseen entities

To better analysis the effectiveness of our model in predicting unseen entities, we not only predict events at time $t+1$ shown in Table 3, but also predict entities at time $t+\Delta t$. The prediction results are shown in Figure 5.

The results show that our model can make effective predictions when faced with unseen entities. Although TiTer performs better when $\Delta t = 1$, the performance of the TiTer model decreases most obviously as Δt increases because TiTer is an inference model based on reinforcement learning and relies heavily on existing facts for reasoning. The TSE of our model also uses recurrent units, so it shows a decrease similar to those of the RE-GCN and RE-NET models. However, because the HSE uses a Hawkes process, its performance decline is significantly slower. Our model combines the TSE and HSE, not only using recurrent units to model the HSE but also using Hawkes processes to capture more long-term information.

Table 9 Case studies of the three levels of SE. We show the most recent four facts in history. Unseen entities and relations are colored in bold. The semantic evidence is marked in italics.

Relation-level SE				
Query	[**Boko Haram**, *Abduct, hijack, or take hostage*, ?, 06/04] → Citizen (Nigeria)			
History	[Armed Rebel (Nigeria), *Abduct* ··· *hostage*, *Citizen (Nigeria)*, 06/03]			
	[Armed Gang (Afghanistan), *Abduct* ··· *hostage*, Employee (India), 06/03]			
	[Militant (Nigeria), *Abduct, hijack, or take hostage*, *Citizen (Nigeria)*, 06/03]			
	[Militant (Boko Haram), *Abduct, hijack* ··· *hostage*, *Citizen (Nigeria)*, 06/03]			
Value	Candidates	p^{ep}_{temp}	p^{ep}_{hist}	p^{ep}_{final}
	Citizen (Nigeria)	3.8757E−3	1.4102E−4	2.7130E−3
	Employee (India)	1.7337E−9	1.3990E−4	1.4069E−4
	...	...	...	...
Entity-level SE				
Query	[*Hamas*, **Use unconventional violence**, *?*, 07/27] → Israeli Defense Forces			
History	[*Hamas*, Make statement, *Israeli Defense Forces*, 07/26]			
	[*Hamas*, Protest violently, riot, *Israeli Defense Forces*,07/25]			
	[*Hamas*, Make statement, *Israeli Defense Forces*, 07/25]			
	[*Hamas*, fight with artillery and tanks, Hezbollah, 07/24]			
Value	Candidates	p^{ep}_{temp}	p^{ep}_{hist}	p^{ep}_{final}
	Israeli Defense Forces	2.3059E−1	1.4363E−4	1.6142E−4
	Hezbollah	3.4335E−8	1.4012E−4	1.4258E−4
	...	...	...	...
Triple-level SE				
Query	[*Citizen (Nigeria)*, *Make an appeal or request*, ?, 08/11] → Government (Nigeria)			
History	[*Citizen (Nigeria)*, *Make an appeal or request*, Government (Nigeria), 08/08]			
	[*Citizen (Nigeria)*, *Make an appeal or request*, Medical Personnel, 08/08]			
	[*Citizen (Nigeria)*, *Make an appeal or request*, Court Judge (Nigeria), 08/07]			
	[*Citizen (Nigeria)*, *Make an appeal or request*, Government (Nigeria), 08/07]			
Value	Candidates	p^{ep}_{temp}	p^{ep}_{hist}	p^{ep}_{final}
	Government (Nigeria)	1.7491E−2	1.4036E−4	1.4274E−4
	Court Judge (Nigeria)	8.0387E−4	1.4030E−4	1.4038E−4
	Medical Personnel (Nigeria)	4.4561E−4	1.4034E−4	1.4033E−4
	...	...	...	...

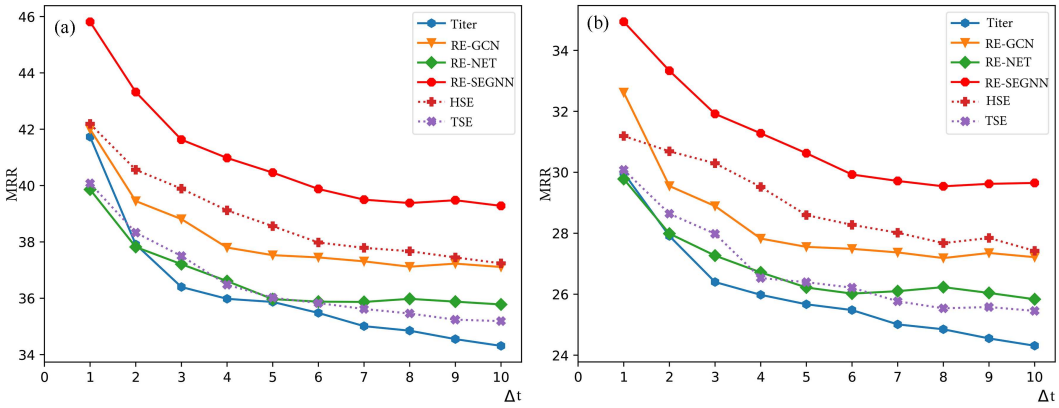

Figure 5 (Color online) MRR of some SOTA models with different Δt. (a) ICEWS14; (b) ICEWS18.

5 Conclusion

We propose the RE-SEGNN model for TKG reasoning with extrapolation setting. The RE-SEGNN uses two types of SE encoder to learn the TSE and HSE, respectively. After combining the score function obtained by the SE-aware decoder, our model can perform better when facing unseen entities and relations while perceiving the impact of historical facts on the current timestamp. Experiments on six public widely used datasets reveal that RE-SEGNN outperforms SOTA models on entity prediction and achieves SOTA

performance on relation prediction. Extensive experiments also validate the effectiveness of each module in our model. Moreover, we show how the TSE and the HSE work when facing unseen entities and relations through case studies.

Acknowledgements This work was supported in part by National Key R&D Program of China (Grant No. 2020AAA0108501) and Major Program (JD) of Hubei Province (Grant No. 2023BAA024).

References

1 Jin W, Qu M, Jin X, et al. Recurrent event network: autoregressive structure inferenceover temporal knowledge graphs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020. 6669–6683
2 Li Z, Jin X, Li W, et al. 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, 2021. 408–417
3 Liu K, Zhao F, Xu G, et al. Temporal knowledge graph reasoning via time-distributed representation Learning. In: Proceedings of the International Conference on Data Mining, Orlando, 2022. 279–288
4 Zhu C, Chen M, Fan C, et al. Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 4732–4740
5 Wu J, Cao M, Cheung J C K, et al. TeMP: temporal message passing for temporal knowledge graph completion. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020. 5730–5746
6 Hawkes A G. Spectra of some self-exciting and mutually exciting point processes. Biometrika, 1971, 58: 83–90
7 Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of the Advances in Neural Information Processing Systems, 2013. 2787–2795
8 Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, Québec, 2014. 1112–1119
9 Nickel M, Tresp V, Kriegel H P. A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, Washington, 2011. 809–816
10 Yang B, Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations, San Diego, 2015
11 Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction. In: Proceedings of the International Conference on Machine Learning, New York City, 2016. 2071–2080
12 Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, Shenzhen, 2018. 1811–1818
13 Vu T, Nguyen T D, Nguyen D Q, et al. A capsule network-based embedding model for knowledge graph completion and search personalization. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, 2019. 2180–2189
14 Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, 2018. 327–333
15 Xiong W, Hoang T, Wang W Y. DeepPath: a reinforcement learning method for knowledge graph reasoning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017. 564–573
16 Yao L, Mao C, Luo Y. KG-BERT: bert for knowledge graph completion. 2019. ArXiv:1909.03193
17 Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, 2019. 4171–4186
18 Wang L, Zhao W, Wei Z, et al. SimKGC: simple contrastive knowledge graph completion with pre-trained language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, 2022. 4281–4294
19 Shen J, Wang C, Gong L, et al. Joint language semantic and structure embedding for knowledge graph completion. In: Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, 2022. 1965–1978
20 Hu Z, Gutiérrez-Basulto V, Xiang Z, et al. Type-aware embeddings for multi-hop reasoning over knowledge graphs. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, Vienna, 2022. 3078–3084
21 Niu G, Li B, Zhang Y, et al. AutoETER: automated entity type representation for knowledge graph embedding. In: Proceedings of the Findings of the Association for Computational Linguistics, 2020. 1172–1181
22 Jiang T, Liu T, Ge T, et al. Encoding temporal information for time-aware link prediction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 2350–2354
23 Dasgupta S S, Ray S N, Talukdar P P. HyTE: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 2001–2011
24 García-Durán A, Dumančić S, Niepert M. Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 4816–4821
25 Lacroix T, Obozinski G, Usunier N. Tensor decompositions for temporal knowledge base completion. In: Proceedings of the International Conference on Learning Representations, Addis Ababa, 2020
26 Trivedi R, Dai H, Wang Y, et al. Know-Evolve: deep temporal reasoning for dynamic knowledge graphs. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 3462–3471
27 Sun H, Zhong J, Ma Y, et al. TimeTraveler: reinforcement learning for temporal knowledge graph forecasting. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Punta Cana, 2021. 8306–8319
28 Li Z, Jin X, Guan S, et al. Search from history and reason for future: two-stage reasoning on temporal knowledge graphs. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021. 4732–4743
29 Han Z, Chen P, Ma Y, et al. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: Proceedings of the International Conference on Learning Representations, 2021
30 Gao Y, Feng L, Kan Z, et al. Modeling precursors for temporal knowledge graph reasoning via auto-encoder structure. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022. 2044–2051
31 Li Y, Sun S, Zhao J. TiRGN: time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, Vienna, 2022. 2152–2158
32 Liu K, Zhao F, Xu G, et al. RETIA: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation. In: Proceedings of the 39th IEEE International Conference on Data Engineering, Anaheim, 2023. 1761–1774
33 Lin Q, Liu J, Mao R, et al. TECHS: temporal logical graph networks for explainable extrapolation reasoning. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, 2023. 1281–1293
34 Liang K, Meng L, Liu M, et al. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, 2023. 1559–1568

35 Sun H, Geng S, Zhong J, et al. Graph Hawkes transformer for extrapolated reasoning on temporal knowledge graphs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, 2022. 7481–7493
36 Zhou K, Zha H, Song L. Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, Atlanta, 2013. 641–649
37 Zhang Q, Lipani A, Kirnap O, et al. Self-attentive Hawkes process. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 11183–11193
38 Zuo S, Jiang H, Li Z, et al. Transformer Hawkes process. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 11692–11702
39 Shang C, Tang Y, Huang J, et al. End-to-end structure-aware convolutional networks for knowledge base completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, 2019. 3060–3067
40 Li R, Cao Y, Zhu Q, et al. How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 5781–5791
41 Li G, Muller M, Thabet A, et al. DeepGCNs: can GCNs go as deep as CNNs? In: Proceedings of International Conference on Computer Vision, Seoul, 2019. 9266–9275
42 Leetaru K, Schrodt P A. GDELT: global data on events, location, and tone, 1979-2012. In: Proceedings of the International Studies Association Annual Convention, San Francisco, 2013. 1–49
43 Leblay J, Chekol M W. Deriving validity time in knowledge graph. In: Proceedings of the Web Conference, Lyon, 2018. 1771–1776
44 Mahdisoltani F, Biega J, Suchanek F. YAGO3: a knowledge base from multilingual Wikipedias. In: Proceedings of the 7th Biennial Conference on Innovative Data Systems Research, Asilomar, 2015
45 Boschee E, Lautenschlager J, O'Brien S, et al. ICEWS coded event data. Harvard Dataverse, 2015, 12: 2
46 Goel R, Kazemi S M, Brubaker M, et al. Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, New York, 2020. 3988–3995
47 Han Z, Ding Z, Ma Y, et al. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Punta Cana, 2021. 8352–8364
48 Han Z, Ma Y, Wang Y, et al. Graph Hawkes neural network for forecasting on temporal knowledge graphs. In: Proceedings of the Conference on Automated Knowledge Base Construction, 2020
49 Dai H L, Peng X, Shi X H, et al. Reveal training performance mystery between TensorFlow and PyTorch in the single GPU environment. Sci China Inf Sci, 2022, 65: 112103