tHR-Net: A Hybrid Reasoning Framework for Temporal Knowledge Graph

Yijing Zhao[1,2], Yumeng Liu[1,2(✉)], Zihang Wan[1], and Hongan Wang[1,2]

[1] Institute of Software Chinese Academy of Sciences, Beijing, China
{yijing,yumeng,hongan}@iscas.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

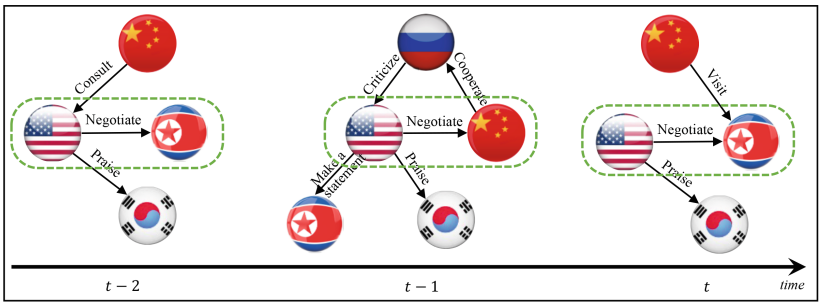
Abstract. Entity prediction and relation prediction are the two major tasks of temporal knowledge graph (TKG) reasoning. The key to answering queries about future events is to understand historical trends and extract the information most likely to affect the future, i.e., the TKG reasoning task is both influenced by the trends of time-evolving graphs and directly driven by the facts relevant to a specific query. Existing methods mostly build models separately for these two characteristics, namely evolution representation learning and query-specific methods, failing to integrate these two crucial factors that determine reasoning results into a single framework. In this paper, we propose a novel temporal hybrid reasoning network (tHR-NET), simultaneously considering the modeling of graph feature space evolution and the enhancement of query-related feature representations in TKG. Specifically, we introduce a global graph space evolution module to extract graph trends, which influence entity/relation representations at each timestamp through a temporal view projection. Additionally, we propose a query-specific increment module for targeted enhancement of entity and relation representations, capturing query-related factors over extended durations. Through extensive experiments on real datasets, tHR-NET demonstrates distinct advantages in parallel entity and relation prediction.

Keywords: Temporal Knowledge Graph Reasoning · Event Prediction · Evolutional Representation Learning · Graph Neural Network

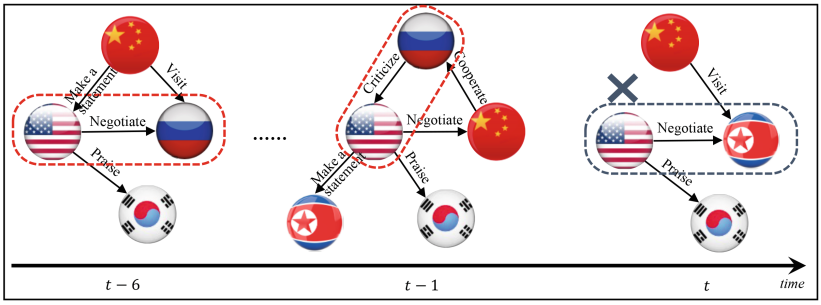
1 Introduction

The temporal knowledge graph (TKG) has been widely employed to model temporal data in various fields such as event reasoning [2,3], resource scheduling [21], and commodity recommendation [19], which brought simplification and efficiency to problem description and solution. On the basis of static KG [1], TKG is composed of a sequence of quadruples, i.e., (subject, relation, object, timestamp), which consists of a complete set of entities and relations existing at one time, organized in chronological order [8]. Due to the incompleteness and dynamic change of graph structure, TKG reasoning is a complex task that worth

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 562, pp. 223–241, 2024.
https://doi.org/10.1007/978-3-031-54528-3_13



(a) Different timestamps *(t-2, t-1, t)* yield varying inference results for the same query *(the United States, Negotiate, ?)* as the TKG evolves.



(b) The events leading to *(the United States, Negotiate, Russia)* at timestamp *t+1* are neither recent nor temporally continuous.

Fig. 1. Two examples of reasoning over TKG.

study. Two kinds of typical issues in TKG reasoning are entity prediction and relation prediction [31], i.e., inferring the entities or relations that will exist in future facts using historical knowledge graph sequences.

Deep learning methods have been developed as one of the main research directions to solve TKG reasoning problems due to its powerful data mining and intelligent computing ability. Formally, this type of methods involves two processes: one is the encoding process, which accurately models and represents input data as low dimensional vectors; the other is the decoding process, which generates problem outputs for downstream tasks based on the represented data. To achieve accurate TKG reasoning for downstream application tasks, there is an increasing need to research representation learning methods to optimize the utilization of historical knowledge graph sequences. Two categories are primarily covered by their optimization directions [1]. One is structural feature extraction, which involves optimizing the fusion and aggregation of entity and relation features within a time snapshot and typically makes use of graph neural networks and their variants [6,12,18]. Another is temporal pattern mining, in which models like recurrent neural networks [3,12,18] or transformers [6] recognize and combine the feature information under various time slices. The combination of

these two categories enables the identification of trends in entity and relation evolution.

The key to answering queries about future events lies in a thorough understanding of history and the ability to extract information most likely to influence the future. Consequently, two principles that need to be adhered to become critical for TKG reasoning: (1) Temporal Variability - graph evolution exhibits continuous variability over time [16]; (2) Causal Discontinuity - A substantial number of events rely on discontinuous histories [6]. In TKG reasoning, the former provides an overall context characterizing the evolution of the knowledge graph, uncovering the variations of the graph over time, and thus influencing local entities and relations. The latter can selectively capture evidence related to specific queries from historical events, thereby reducing the interference from non-causal events in continuous time. Figure 1 illustrates these two phenomena, respectively. In Fig. 1a, as the graph evolves, the same prediction task under different timestamps *(t-2, t-1, t)*-for example, the query *(the United States, Negotiate, ?)*-yields different inference results. In Fig. 1b, the events causing the occurrence of the event *(the United States, Negotiate, Russia)* at timestamp *t+1* are neither the most recent nor temporally continuous.

Existing models have often unilaterally focused on one of these two kinds of problems, and can thus be categorized into: evolution representation learning methods and query-specific methods [15]. Evolution representation learning methods [15,18,27,29] are based on updating the embeddings of each entity and relation at each timestamp within historical KGs, learning the trends of feature evolution. Query-specific methods [6,12] retrieve context information from historical KGs for a query, such as entities and relations. These kinds of methods possess advantages such as low space occupancy and rapid convergence, but they overlook the impact of global graph feature space evolution on entity and relation representation. HisMatch [16] considers both candidate and query histories, but its use of recurrent neural networks in extracting features from query-related historical events leads to struggles with temporal forgetting, thereby hindering the extraction of relevant events.

In this paper, we propose a novel temporal hybrid reasoning network, called tHR-NET, that simultaneously considers the aforementioned two issues by unifying the modeling of graph feature space evolution and the enhancement of query-relevant feature representations in TKG. Specifically, we propose a global graph space evolution module in order to effectively extract graph evolution trends which will influence the entity/relation representations at every timestamp. Meanwhile, we design a local feature aggregation module to acquire entity/relation representations. Following that, we apply a computation with temporal view projection to align the aggregated local entities/relations features with the current graph evolution features, ensuring adherence to the temporal variability inherent in graph evolution. Then, we propose the query-specific increment module to enhance the entity/relation representations based on specific queries, enabling the capture of factors most relevant to the queries over a long duration. Finally, the entity representations and relation representations are obtained in parallel, enabling entity and relation prediction tasks by connecting

any downstream application task model. Our contributions can be summarized as follows:

- We designed a novel temporal hybrid reasoning network framework. To the best of our knowledge, this is the first study that simultaneously models both the continuous variability of graph evolution over time and the non-continuous features of query-related history in TKG reasoning.
- We propose a new temporal projection technique to align the aggregated local features of entities/relations with the current graph evolution feature, which ensures entity/relation representations adherence to the continuous variability inherent in graph evolution.
- We have developed a query-specific increment module, capable of caching and extracting features from the representations output by entity/relation evolution learning based on specific queries, which can break the temporal forgetfulness in evolution representation learning.
- Extensive experiments demonstrate that tHR-NET achieves significantly better performance over both entity and relation prediction tasks on four benchmarks by modeling the hybrid reasoning framework comprehensively.

2 Related Work

Temporal knowledge graph (TKG) reasoning settings encompass both interpolation and extrapolation [12]. Interpolation approaches predominantly focus on inferring and completing absent information within historical time points. Typical techniques such as HyTE [4], TTransE [13], and TA-DistMult [7] have been employed, while their capacity to foresee future occurrences remains limited. Extrapolation methodologies deduce novel facts, encompassing subjects, objects, and relations, that may transpire in the future by relying on a chronologically arranged series of historical facts. By capturing historical event information, these methods seek to make a reasonable prediction of the future, which has recently become a research hotspot. They can be divided into two classes based on how they use the events historical information: evolution representation learning and query-specific methods [27].

Evolution Representation Learning Methods. These methods are based on encoding each entity and relation at each timestamp within historical KGs, learning the trends of feature evolution, and considering the query only at the decoding stage. For example, RE-GCN [12] encodes fixed-length historical facts into the entity and relation evolutionary representation for future fact prediction. CEN [15] models variable-length facts sequences. TANGO [10] employs neural ordinary differential equations to model the structural information for each candidate entity. CRNet [27] effectively models the mutual influence of events that occur concurrently in the future.

Query-Specific Methods. These methods retrieve context information related to the query from historical KGs. RE-NET [12] employs both GCN and GRU to model the sequence of 1-hop subgraphs associated with the given subject entity.

xERTE [9] learns a dynamic pruning process to identify subgraphs related to the query. Both CluSTeR [17] and TITer [25] utilize reinforcement learning to discover paths relevant to the query in history. rGalt [6] augments the capacity to extract features of interest over extended periods by embedding graphs at distinct time points into parallel input transformer structures.

The recent method HiSMatch [16] considers the query-related histories and evolution histories of all the candidate entities under the matching framework. But its use of recurrent neural networks in extracting features from query-related historical events leads to struggles with temporal forgetting.

3 Preliminaries

A TKG $\mathcal{G}$ can be formalized as a sequence of KGs with timestamps, i.e., $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_t, ...\}$. Let $\mathcal{E}$ be a finite set of entities and $\mathcal{R}$ be a finite set of relations belong to $\mathcal{G}$. The KG of each time t can be represented as a directed multi-relational graph $\mathcal{G}_t = (\mathcal{E}_t, \mathcal{R}_t)$, where $\mathcal{E}_t \in \mathcal{E}$ is the set of entities and $\mathcal{R}_t \in \mathcal{R}$ is the set of relations at timestamp t. Any fact in $\mathcal{G}_t$ can be denoted as a triplet, (s_t, r_t, o_t), which represents a fact of relation $r_t \in \mathcal{R}_t$ that occurs with $s_t \in \mathcal{E}_t$ as its subject entity and $o_t \in \mathcal{E}_t$ as its object entity at timestamp t. Under the assumption that the facts predictions at a future timestamp t+1 depends on the KGs at the latest k timestamps (i.e., $\{\mathcal{G}_{t-k+1}, ..., \mathcal{G}_{t-1}, \mathcal{G}_t\}$), the two temporal prediction problems can be formulated as follows:

Problem 1. Entity Prediction. The entity prediction task aims to predict the missing object entity of a query $(s_{t+1}, r_{t+1}, ?_{t+1})$ and the missing subject entity of a query $(?_{t+1}, r_{t+1}, o_{t+1})$ which can be converted to $\left(s_{t+1}, r_{t+1}^{-1}, ?_{t+1}\right)$. We model the conditional probability vector of all object entities with the subject entity s, the relation r and the historical KG sequence $\mathcal{G}_{t-k+1:t}$:

$$\mathbb{P}\left(o|s, r, \mathcal{G}_{t-k+1:t}\right) \tag{1}$$

Problem 2. Relation Prediction. The relation prediction task attempts to predict the missing relation of a query $(s_{t+1}, ?_{t+1}, o_{t+1})$. We model the conditional probability vector of all relations with the subject entity s, the object entity o and the historical KG sequence $\mathcal{G}_{t-k+1:t}$:

$$\mathbb{P}\left(r|s, o, \mathcal{G}_{t-k+1:t}\right) \tag{2}$$

4 Methodology

We introduce a novel **t**emporal **H**ybrid **R**easoning **Net**works (tHR-NET) that unifies the concepts of evolution representation learning and query-specific methods to perform reasoning over the TKG. Figure 2 illustrates the working flow of tHR-NET, which comprises four key modules: 1) Global Graph Space Evolution - This module performs global graph features vectorization at each time and

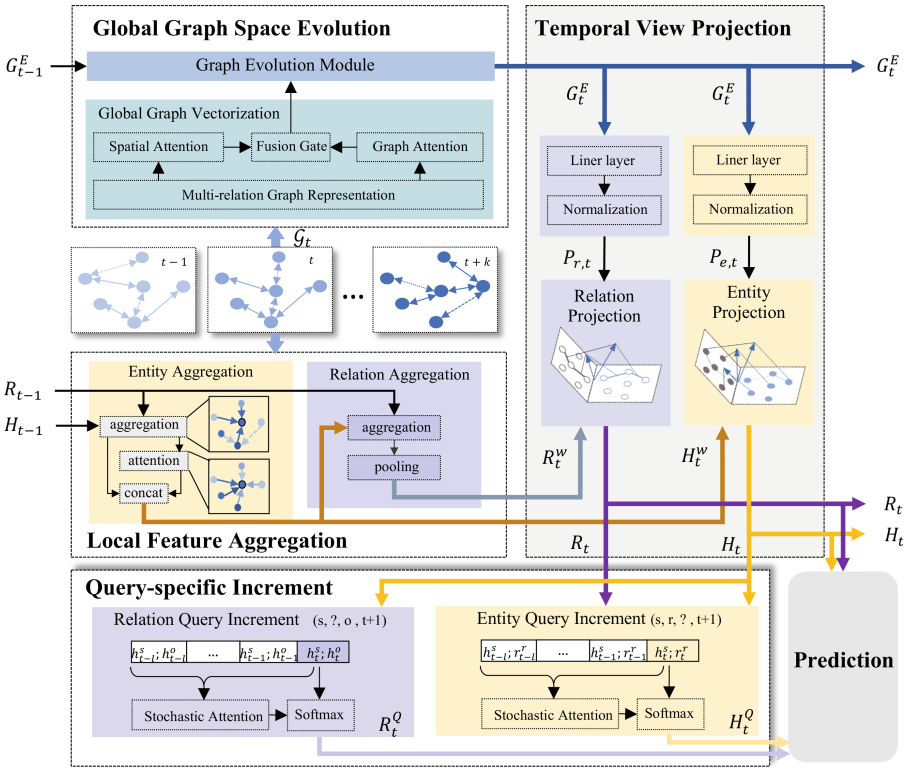


Fig. 2. An illustrative diagram of tHR-NET for TKG reasoning at timestamp t.

then obtains the global graph evolution representation. 2) Local Feature Aggregation - It aggregates entity and relation features through the known structural information and uncertain attention information from the micro-level perspective. 3) Temporal View Projection - It aligns the aggregated local features of entity/relation representations to the current graph evolution space through the projection technique. 4) Query-specific Increment - This module enhances the entity/relation representations based on specific queries, enabling the capture of factors most relevant to the query over a long duration. In the following sections, we provide detailed explanations of each of the four modules.

4.1 Global Graph Space Evolution

This module focuses on the representation of global graph feature evolution trends over time, which provides a macro-level alignment and adjustment background for entity and relation evolution.

Global Graph Vectorization. To form complete graph feature vectorization at each time, tHR-NET extracts the interactions between entities from different

relational perspectives. We utilize matrix $G^M_{r_i}(t) \in \mathbb{R}^{N\times N}$ as one relation graph and each element represents whether exists a relation r_i between entities in $\mathcal{G}_t$. Then we employ a fully-connected neural network to transform the matrix into a multidimensional vector $G^M_{r_i,e_j}(t) \in \mathbb{R}^H$, where r_i, e_j is an arbitrary relation and entity at any time.

In order to acquire inner and outer correlations of multi-relation graphs, we design a multi-layer spatial-graph attention fusion component. The l-th layer input is denoted as $G^{(l-1)}$, and the hidden state of entity e_j on relation graph r_i is denoted as $g^{(l-1)}_{r_i,e_j}$. The 1-th layer input is initialized as $g^{(0)}_{r_i,e_j} = G^M_{r_i,e_j}$. Then, spatial attention and graph attention calculations are performed at each layer.

As an illustration, consider the graph attention mechanism, which captures the correlations between the same entities in multi-relation graphs. The h-th head graph attention coefficient $\beta^{(h)}_{r_i,r_m}$ is calculated as follows:

$$\beta^{(h)}_{r_i,r_m} = \frac{exp\left(< f^{(h)}_g\left(g^{(l-1)}_{r_i,e_j}\right), f^{(h)}_g\left(g^{(l-1)}_{r_m,e_j}\right) >\right)}{\sum_{m\in M} exp\left(< f^{(h)}_g\left(g^{(l-1)}_{r_i,e_j}\right), f^{(h)}_g\left(g^{(l-1)}_{r_m,e_j}\right) >\right)} \tag{3}$$

where $f^{(h)}_g$ is ReLU function, and r_m can be an arbitrary relation graph. Then we obtain the next layer state from graph attention for e_j in relation r_i as follows:

$$g^{g,(l)}_{r_i,e_j} = \sum_{h=1}^{H}\left(\sum_{m=1}^{M}\beta^{(h)}_{r_i,r_m}\cdot f^{(h)}_g\left(g^{(l-1)}_{r_m,e_j}\right)\right) \tag{4}$$

The spatial attention mechanism mainly captures the correlations of different entities in the same relation graph. Similarly, we can calculate the spatial attention coefficient $\alpha^{(h)}_{e_j,e_n}$ and the next layer state $g^{s,(l)}_{r_i,e_j}$.

To further combine correlations on different spatial and graphs, outputs from spatial attention $G^{s,(l)}$ and graph attention $G^{g,(l)}$ in the l-th block are fused:

$$G^{(l)} = \omega \otimes G^{s,(l)} + (1-\omega)\otimes G^{g,(l)} \tag{5}$$

where ω is a gate inspired by [30] and $\otimes$ indicates a Hadamard product operation.

Graph Evolution Module. We update the global graph evolution embedding matrix G^E_{t-1} to G^E_t via the GRU with the fusion feature G_t obtained before:

$$G^E_t = GRU\left(G^E_{t-1}, G_t\right) \tag{6}$$

where G^E_t are randomly initialized at the time of $t = 0$.

4.2 Local Feature Aggregation

This module focuses on micro-level entity and relation feature extraction, which utilizes local structure information and mines hidden information via the attention mechanism.

Entity Feature Aggregation. In order to acquire complete and accuracy entity feature representations, tHR-NET considers the impacts of both known local structure and uncertain attention on entities. To this end, we utilize a multi-layer GCN for entity feature aggregation and obtain the local information in the l-th layer as follows:

$$h_{o,t}^{(l),loc} = f\left(\sum_{(s,r),\exists(s,r,o)\in\mathcal{G}_t} W_l^1\left(h_{s,t}^{w,(l-1)} + r_{t-1}\right) + W_l^2 h_{o,t}^{w,(l-1)}\right) \tag{7}$$

where $h_{s,t}^{w,(l-1)}$,$h_{o,t}^{w,(l-1)}$ are the elements in $H_t^{w,(l-1)}$, r_{t-1} is the element in R_{t-1}. W_l^1 and W_l^2 are the learnable parameters, and $f(\cdot)$ is ReLU function. The 1-th layer input $H_t^{w,(0)} = H_{t-1}$. Inspired by [2], a structure-aware Transformer can be constructed to consolidate the attention information concerning the entities.

$$h_{v,t}^{(l),nloc} = h_{v,t}^{(l),loc} + 1/\sqrt{d_v} \cdot \sum_{u,w\in\mathcal{N}_k(v)} \frac{exp(< h_{v,t}^{(l),loc}, h_{u,t}^{(l),loc} >)}{\sum exp(< h_{v,t}^{(l),loc}, h_{w,t}^{(l),loc} >)} h_{u,t}^{(l),loc} \tag{8}$$

where d_v denotes the number of the k-hop neighborhood $\mathcal{N}_k(v)$ of entity v, and u, w are the entities within $\mathcal{N}_k(v)$. k is an adjustable hyperparameter and the larger k is, the more uncertain information is concerned. Especially, to combine the known structural information with the uncertain attention information of entities, we utilize the residual technique by concatenating both types of information and passing through a linear layer to produce the output $h_t^{w,(l)}$. The final embedding matrix H_t^w can be obtained after several aggregations.

Relation Feature Aggregation. Similar to entity aggregation, the calculation of relation aggregation is also directional. For each relation $r \in \mathcal{R}$ existing in $\mathcal{G}_t$ at time t, the update is calculated as follows:

$$r_t^w = f\left(\sum_{(s,o),\exists(s,r,o)\in\mathcal{G}_t} W_r^1\left(h_{o,t}^w - h_{s,t}^w\right) + W_r^2 r_{t-1}\right) \tag{9}$$

where W_r^1 and W_r^2 are the learnable parameters, and $f(\cdot)$ is ReLU function. Then we can acquire relation embedding matrix R_t^w.

4.3 Temporal View Projection

We apply a temporal space projection computation to align the aggregated local features of entity/relation representations with the current graph evolution space. This ensures that the entities/relations representations after local feature aggregation also adhere to the continuous variability inherent in the graph evolution.

To capture projection space, graph evolution embedding G_{t-1}^E and G_t^E are used calculating the evolution dynamics of the entity and relation $P_{e,t}$ and $P_{r,t}$:

$$P_{e,t} = W_p^1 (G_{t-1}^E)^{-1} G_t^E + b_p^1 \tag{10}$$

$$P_{r,t} = W_p^2 (G_{t-1}^E)^{-1} G_t^E + b_p^2 \tag{11}$$

where W_p^1, W_p^2, and b_p^1, b_p^2 are learnable parameters, which represent entities and relations will be projected into different parameter spaces. Then we restrict $||P_{e,t}||_2 = 1$ and $||P_{r,t}||_2 = 1$ by normalization. After that, we use the entity embedding H_t^w and relation embedding R_t^w calculating the projection [4] to transform their feature space into current feature space. Each entity and relation evolution embedding h_t and r_t are calculated by different learnable weights as follows:

$$h_t = h_t^w + (W_p^3 P_{e,t})^T h_t^w \left(W_p^3 P_{e,t}\right) \tag{12}$$

$$r_t = r_t^w + (W_p^4 P_{r,t})^T r_t^w \left(W_p^4 P_{r,t}\right) \tag{13}$$

Finally, we acquire entity and relation evolution embedding matrices H_t and R_t.

4.4 Query-Specific Increment

This module aims to break the temporal forgetfulness in the evolution representation of entities and relations, thereby reducing or eliminating the reliance on recent non-causal related events in inference. It enhances the entity/relation representations based on specific queries, enabling the capture of aspects most relevant to the query over a long duration.

We designed a query-based buffer pool that can retain information related to queries over an extended time period l. Taking the entity prediction query $(s, r, ?, t+1)$ as an example, the information in the query buffer pool can be represented as:

$$X_{(s,r),t} = \{(h_{t-l}^s; r_{t-l}^r), ..., (h_{t-1}^s; r_{t-1}^r), (h_t^s; r_t^r)\} \tag{14}$$

where $h_k^s, r_k^r,\ k \in [t-l,\ ..., t-1, t]$ are the vector representations of entity s and relation r from H_t and R_t respectively. Similarly, for the relation prediction query $(s, ?, o, t+1)$, the result is $X_{(s,o),t}$.

To minimize the noise impact of non-causal factors and retain only the most relevant historical aspects, we make improvements to a type of random attention mechanism [22]. For each timestamp information in the query buffer pool $(h_k^s; r_k^r)$, we map it to $p_{(s,r),k} \in [0, 1]$ through an MLP layer followed by a sigmoid activation function. Then, for each forward training, we draw random attention from the Bernoulli distribution $\alpha_{(s,r),k} \sim Bern(p_{(s,r),k})$. To ensure that the gradient is computable, we apply the gumbel-softmax reparameterization trick [11]. Subsequently, we obtain the information increment corresponding to the entity prediction query as follows:

$$(h_t^{Q,s},\ r_t^{Q,r}) = \sum_{k=t-l}^{l} \alpha_{(s,r),k} \cdot X_{(s,r),k} \tag{15}$$

Similarly, for relation prediction queries, the information increment can be obtained as $(h_t^{Q,s},\ h_t^{Q,o})$. To facilitate a unified representation, we denote the collection of the above entity and relation enhanced representations as H_t^Q and R_t^Q, respectively.

4.5 Prediction and Loss Function

This model is to solve the entity and relation prediction problems in TKG reasoning. We choose ConvTransE [24] as the downstream task decoder and acquire the probability vector of all entities and relations:

$$p\left(o|s,r,H_t,R_t\right)=\sigma\left(H_t ConvTransE\left(s_t+s_t^Q,r_t+r_t^Q\right)\right) \tag{16}$$

$$p\left(r|s,o,H_t,R_t\right)=\sigma\left(R_t ConvTransE\left(s_t+s_t^Q,o_t+o_t^Q\right)\right) \tag{17}$$

where $\sigma(\cdot)$ is the sigmoid function, s_t, r_t, o_t, s_t^Q, r_t^Q, o_t^Q are the embeddings of s, r and o at timestamp t in H_t, R_t, H_t^Q and R_t^Qrespectively.

We construct the objective function to guide both the evolution learning of entities/relations and the query increment to produce the desired effects during training our model. Formally, we use a binary crossentropy loss to optimize the model parameters as follows:

$$L_e=\sum_{t=0}^{T-1}\sum_{(s,r,o)\in\ G_{t+1}}\left(-log\ p\left(o\,|s,r,H_t,R_t\right)-\beta\cdot log\ \left(1-p\left(o\,|s,r,H_t^w,R_t^w\right)\right)\right) \tag{18}$$

$$L_r=\sum_{t=0}^{T-1}\sum_{(s,r,o)\in\ G_{t+1}}\left(-log\ p\left(r\,|s,o,H_t,R_t\right)-\beta\cdot log\ \left(1-p\left(r\,|s,o,H_t^w,R_t^w\right)\right)\right) \tag{19}$$

where T is the number of timestamps in the training set. While $p\left(o|s,r,H_t^w,R_t^w\right)$ and $p\left(r|s,o,H_t^w,R_t^w\right)$ without projection are regarded as negative samples to accelerate convergence process. β is the hyper-parameter to balance positive and negative samples.

In order to combined different prediction task loss, we minimize the following objective function to learn the model parameter:

$$L=\alpha L_e+(1-\alpha)\,L_r \tag{20}$$

where α is the hyper-parameter to balance the weight of the multi-task learning problem.

5 Experiment

In this section, we perform experiments to validate the effectiveness of our proposed tHR-NET model by investigating the research questions as follows:

RQ1: How does our proposed tHR-NET method perform compared with state-of-the-art TKG facts prediction methods?

RQ2: How do the different components of tHR-NET contribute to the model performance?

RQ3: How do the configurations of model hyperparameters affect the prediction performance?

RQ4: Can the proposed tHR-NET model show interpretability, through the effective modeling of TKGs evolution trends and the enhancement of query-related features?

5.1 Experimental Settings

Datasets. The experimental evaluation was executed on four different event datasets with three different lengths of time intervals. Table 1 provides a summary of the dataset statistics. The **ICEWS14** and **ICEWS05-15** datasets [7] originate from the Integrated Crisis Early Warning System (ICEWS), a repository containing political events, which occurred within a time interval of 24 h. **YAGO** [20], is a comprehensive semantic knowledge base from wiki resources, and our experiments are taken from a subset of it with a 1-year time interval. **GDELT** [14] is drawn from the Global Database of Events, Language, and Tone with a 15-minute time interval. Each of these aforementioned datasets was partitioned into training, validation, and test sets, approximately maintaining a proportion of 8:1:1 respectively.

Table 1. Dataset Statistics.$|\mathcal{S}_{train}|, |\mathcal{S}_{valid}, |\mathcal{S}_{test}|$ are the numbers of facts in training, validation, and test sets. The time interval donates time granularity between temporally adjacent facts.

Datasets	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{S}_{train}\rvert$	$\lvert\mathcal{S}_{valid}\rvert$	$\lvert\mathcal{S}_{test}\rvert$	time interval
ICEWS14	6869	230	74,845	8,514	7,371	24 h
ICEWS05-15	10,094	251	368,868	46,302	46,159	24 h
YAGO	10,623	10	161,540	19,523	20,026	1 year
GDELT	7,691	240	1,734,399	238,765	305,241	15 min

Baselines. To fully evaluate the performance of our method, we compare tHR-NET with static KG embedding models and TKG embedding models as well as state-of-the-art methods. For **entity prediction**, we opt for DistMult [28], ComplEx [26], R-GCN [23], ConvE [5], and ConvTransE [24] as static models, disregarding temporal data to facilitate the completion of absent entities within knowledge graphs. We employ HyTE [4], TTransE [13], TA-DistMult [7], RE-NET [12], RE-GCN [18], rGalT [6], CEN [15], HisMatch [16] and CRNet [27] as TKG embedding models. For **relation prediction**, we choose from the aforementioned models that are directly applicable to address relation completion or prediction challenges including ConvE [5] and ConvTransE [24] for static models, and RE-GCN [18] for temporal models.

Evaluation Metrics. To evaluate our method quantitatively, we adopt two widely used metrics for entity prediction and relation prediction [6,12,18]: ***MRR*** denotes the Mean Reciprocal Ranks which is a more stable evaluation metric compared with MR [12]. ***Hits@k*** denotes the percentage that the top-k recommended list contains at least one ground truth item. In our experiments, k is set to 1, 3 and 10. Specifically, for the entity prediction task on YAGO, since the previous works RE-NET and RE-GCN did not report the results of $Hits@1$, we then only report the results of MRR and $Hits@\{3,\ 10\}$. Following previous work [6], during evaluations, We compare the experimental results under the raw setting rather than the filtered setting utilized in [12].

Implementation Details. We implement our model using Pytorch and adopt Adam optimizer for model optimization. During the training phase, the model inference process is conducted with a learning rate of 0.001 configured with a 0.99 decay rate. The default setting for hidden state dimensionality is 100 and the dropout rate is 0.5. The length of historical sequences in the Global Graph Space Evolution module is chosen from the range [1, 2, 3, 4, 5, 6] with the best setting of 4 for ICEWS14 and ICEWS05-15 dataset, 3 for YAGO dataset and 5 for GDELT dataset. The number of entity and relation aggregation layers is set to 2 and 1 for all datasets respectively. The length of the extended time period in the Query-specific Increment module is set to 20 for all datasets. For the downstream ConvTransE model, we refer to the settings of RE-GCN [18]. The hyperparameter α for balancing the multi-task learning problem, and the hyperparameter β for balancing the two representation learning process, are optimized according to grid search in different datasets.

Table 2. Performance results of the proposed method vs. baselines for the entity prediction task over all datasets (%).

Model	ICEWS14				ICEWS05-15				YAGO			GDELT			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@3	H@10	MRR	H@1	H@3	H@10
DistMult	20.32	6.13	27.59	46.61	19.91	5.63	27.22	47.33	44.05	49.70	59.94	8.61	3.91	8.27	17.04
ComplEx	22.61	9.88	28.93	47.57	20.26	6.67	26.43	47.31	44.09	49.57	59.64	9.84	5.17	9.58	18.23
R-GCN	28.03	19.42	31.95	44.83	27.13	18.83	30.41	43.16	20.25	24.01	37.30	12.17	7.40	12.37	20.63
ConvE	30.30	21.30	34.42	47.89	31.40	21.56	35.70	50.96	41.22	47.03	59.90	18.37	11.29	19.36	32.13
ConvTransE	31.50	22.46	34.98	50.03	30.28	20.79	33.80	49.95	46.67	52.22	62.52	19.07	11.85	20.32	33.14
HyTE	16.78	2.13	24.84	43.94	16.05	6.53	20.20	34.72	14.42	39.73	46.98	6.69	0.01	7.57	19.06
TTransE	12.86	3.14	15.72	33.65	16.53	5.51	20.77	39.26	26.10	36.28	47.73	5.53	0.46	4.97	15.37
TA-DistMult	26.22	16.83	29.72	45.23	27.51	17.57	31.46	47.32	44.98	50.64	61.11	10.34	4.44	10.44	21.63
RE-NET	35.77	25.99	40.10	54.87	36.86	26.24	41.85	57.60	46.81	52.71	61.93	19.60	12.03	20.56	33.89
RE-GCN	40.39	30.66	44.96	59.21	48.03	37.33	53.85	68.27	58.27	65.62	75.94	19.64	12.42	20.90	33.69
rGalT	38.33	28.57	42.86	58.13	38.89	27.58	44.19	59.10	51.54	54.76	68.31	19.56	12.11	20.89	34.15
CEN	42.20	30.08	47.46	61.31	43.76	31.58	50.10	63.77	54.92	55.77	71.04	21.16	13.43	22.71	36.38
CRNet	48.37	38.21	53.79	67.79	48.90	39.14	54.93	69.18	60.65	63.98	74.62	**25.32**	**15.39**	**27.82**	**44.07**
HisMatch	46.42	35.91	51.63	66.84	52.85	42.01	**59.05**	**73.28**	59.97	64.08	74.35	22.01	14.45	23.80	36.61
tHR-Net	**51.74**	**39.52**	**55.08**	**70.53**	**52.97**	**43.12**	58.84	72.97	**69.54**	**71.54**	**88.78**	23.00	15.19	26.84	40.03

Table 3. Performance results of the proposed method vs. baselines for the relation prediction task over all datasets (%).

Model	ICEWS14	ICEWS05-15	YAGO	GDELT
ConvE	38.80	37.89	91.33	18.84
ConvTransE	38.40	38.26	90.98	18.97
RE-GCN	39.73	38.56	95.18	19.17
tHR-NET	**50.78**	**51.05**	**97.87**	**22.39**

5.2 Performance Results (RQ1)

Results on Entity Prediction. Table 2 presents a comprehensive comparison between our proposed method and baseline approaches for entity prediction tasks on four TKG datasets: ICEWS14, ICEWS05-15, YAGO, and GDELT. tHR-NET consistently achieves the best or near-best results on the aforementioned datasets. Specifically, tHR-NET performs much better than all static methods (the first block) and temporal models under interpolation settings (the second block) primarily due to our model considers temporal factors and has abilities to predict.

Concerning temporal models based on extrapolation settings (the third block), tHR-NET outperforms query-specific methods RE-NET and rGalT on all datasets. Unlike those methods, which focus solely on historical facts related to entities, tHR-NET also considers the possibility of future events that have not yet occurred. This makes tHR-NET more adept at mining and exploring events, leading to its superior performance. Compared to query-specific methods, evolution representation learning methods such as RE-GCN, CEN, and CRNet more rationally model the continuous process of TKG evolving over time, achieving superior results. Specifically, CRNet, through a post-processing module predicting interactions of concurrent related events, obtains the best results on the GDELT dataset, where numerous concurrent events exist. HisMatch considers both evolution and query-related histories, but its reliance on recurrent neural networks for extracting features from query-related historical events results in challenges with temporal forgetting. In contrast, our tHR-NET's query-specific increment module has the ability to capture the most relevant information to a query over extended periods, achieving better results in most cases.

Results on Relation Prediction. In Table 3, we present the experimental results of the MRR metric comparison, which show evidence that tHR-NET surpasses all other baselines in all the datasets. This superior performance can probably be attributed to our model's ability to catch both the global evolution trends and the query-related information, which provides accurate modeling for relation representation and reasoning.

Table 4. Ablation studies on entity prediction.

Model	ICEWS14	ICEWS05-15	YAGO	GDELT
tHR-Net	**51.74**	**52.97**	**69.54**	**23.00**
- *GGSE*	37.16	39.21	45.74	16.16
+ *FCN*	42.89	44.78	51.34	19.27
- *local agg.*	45.08	47.12	58.57	19.97
- *nlocal agg.*	47.31	47.04	60.01	20.64
- *PW*	43.17	46.57	60.22	21.01
- *Lw loss*	48.53	44.60	59.19	21.76
- *query inc.*	39.55	39.88	64.39	19.94

Table 5. Ablation studies on relation prediction.

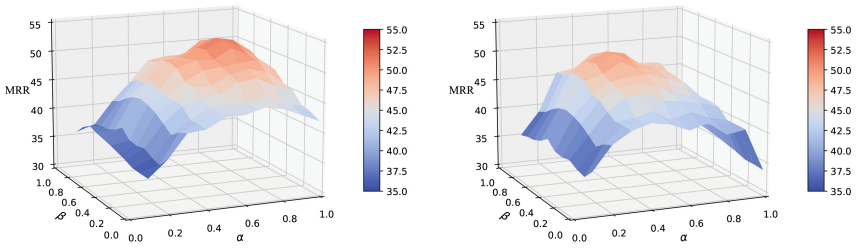
Model	ICEWS14	ICEWS05-15	YAGO	GDELT
tHR-NET	**50.78**	**51.05**	**97.87**	**22.39**
- *GGSE*	40.09	40.78	94.00	18.29
+ *FCN*	45.92	45.30	95.32	19.07
- *local agg.*	47.03	48.03	96.19	19.39
- *nlocal agg.*	46.58	47.91	96.99	19.16
- *PW*	48.12	46.58	96.04	19.88
- *Lw loss*	47.65	47.42	96.72	20.75
- *query inc.*	41.97	42.08	97.85	19.76

5.3 Ablation Study (RQ2)

To verify the effects of the designed different components in our model, we conduct ablation experiments on the above datasets. We discuss the effects of each component as follows:

Effect of Global Graph Space Evolution. As demonstrated in Tables 4 and 5, we removed the Global Graph Space Evolution module, denoted as "-*GGSE*". The term "+*FCN*" indicates the replacement of the Spatial-Graph Attention Fusion method with a simple single-layer fully connected network. A notable impact on the results for all datasets can be observed, signifying that this module enables the model to capture the evolutionary trends and preferences in the global feature space of graphs.

Effect of Local Feature Aggregation. The "-*local agg.*" scenario represents the removal of the local structure aggregation layer based on GCN from the Local Feature Aggregation module. Only the attention mechanism based on Transformers is retained. Conversely, the "-*nlocal agg.*" scenario is the exact opposite. The results reveal that both two feature aggregation mechanisms effectively impact the model's performance.



(a) Entity Prediction (b) Relation Prediction

Fig. 3. Parameter sensitivity analysis on ICEWS14 dataset.

Effect of Temporal Evolution Learning. We modified Eq. (8) and investigated the use of the same projection matrix for entity and relation calculation in the Temporal View Projection module over time, denoted as "*-PW*". It was observed that the performance of the "*-PW*" scenario slightly declined across all four datasets. Additionally, the "*-Lw loss*" scenario retains only the loss functions for entity and relation prediction tasks. The performance demonstrated the important role of evolution negative samples in facilitating model convergence.

Effect of Query-Specific Increment. By removing the Query-specific Increment module, denoted as "*-query inc.*", we observed a notable impact on the results across all datasets. This phenomenon indicates that selectively relying on query-related history, rather than strictly depending on recent history, can lead to more accurate representations of entities and relations.

5.4 Parameter Sensitive (RQ3)

Figure 3 displays the influence of the hyperparameters α and β on entity prediction and relation prediction performance on the ICEWS14 dataset. α balances the entity prediction with the relation prediction results, while β balances downstream tasks with evolution trends. As observed, within the α range of 0.5 to 0.9, there is an inverse relation between the entity and relation predictions changes trends consistent with intuitive expectations. At approximately 0.8, β yields relatively better results for both tasks.

5.5 Case Study (RQ4)

This section provides an explanation for the model's predictions by examining the trends in the changes of inter-entity correlations during the graph evolution process. We investigated a selected query *(Catherine Ashton, Express intent to meet or negotiate, ?, 339)* from the ICEWS14 test set. Figuer 4 shows how tHR-NET infers the prediction result for this query by mining the trend of correlation changes between entities throughout the evolution process of relevant facts. Here,

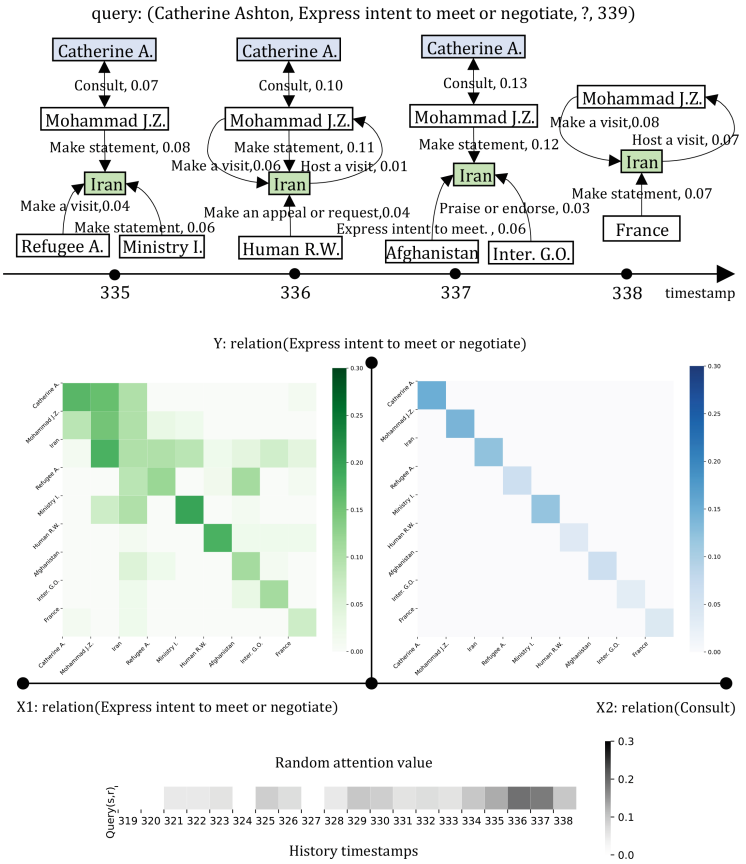


Fig. 4. An example to illustrate how tHR-NET identifies the evolution trend of facts. The upper part displays the subgraphs of facts related to the query *(Catherine Ashton, Express intent to meet or negotiate, ?, 339)* at different timestamps. The middle part presents the two types of attention coefficients of the global graph evolution module for relevant entities at timestamp 339. The lower part presents the random attention value for the recent 20 timestamps.

the historical time refers to timestamps from 335 to 338. The upper part of Fig. 4 displays the subgraphs consisting of partially relevant facts at different timestamps. We model the attention scores for associated entities of a given entity type using Eq. (8) to quantify the relevance between entities. It can be observed that the correlation between the same two entities changes according to a certain trend as time progresses. For instance, as *(Catherine Ashton, Consult, Mohammad Javad Zarif)* repeatedly appears at timestamps 335, 336, and 337, the correlation between the two entities gradually strengthens, resulting in a higher attention coefficient, which is consistent with human reasoning intuition. The middle part of Fig. 4 shows the two types of attention coefficients in the

global graph evolution module for relevant entities at timestamp 339. The left graph displays the spatial attention coefficients calculated by Eq. (3) for entities in the subgraph of the relation *Express intent to meet or negotiate*. The right graph presents the graph attention coefficients between the same entities in the subgraphs of relations *Express intent to meet or negotiate* and *Consult*. We can observe that the entities *Catherine Ashton*, *Mohammad Javad Zarif* and *Iran* demonstrate notable associations in both perspectives, which can be regarded as the inference evolution trend for this query. The lower part of Fig. 4 shows the random attention values in the query-specific increment module for the recent 20 timestamps. We can observe that event reasoning can be influenced by features over an extended period without apparent temporal forgetfulness. At timestamps 336 and 337, higher values indicate the capture of influential factors related to the query, reflecting the model's ability to retain relevant historical information.

6 Conclusion

We proposed a novel temporal hybrid reasoning network for TKG reasoning, referred to as tHR-NET, which simultaneously considers the continuous variability of graph evolution and the causal discontinuity of query-related histories for the first time. Our proposed temporal projection technique aligns local entities/relations representations with graph evolution dynamics, maintaining consistency with temporal continuous variability. Additionally, the query-specific increment module addresses the issue of temporal forgetfulness in evolution representation learning, leading to improved reasoning performance. Experimental results on four commonly used datasets demonstrate that our model exhibits excellent performance in both entity prediction and relation prediction tasks. Our future work aims to introduce the idea of the information bottleneck to further refine the representation of query-related histories.

Acknowledgements. The authors would like to thank the anonymous reviewers for their valuable comments. This work is supported by National Key Research and Development Program of China No. 2022-JCJQ-JJ-0587.

References

1. Barros, C.D.T., Mendonça, M.R.F., Vieira, A.B., Ziviani, A.: A survey on embedding dynamic graphs. ACM Comput. Surv. **55**(1), 1–37 (2023)
2. Chen, D., O'Bray, L., Borgwardt, K.: Structure-aware transformer for graph representation learning. In: Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 3469–3489 (2022)
3. Chen, J., Wang, X., Xu, X.: GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction. Appl. Intell. **52**(7), 7513–7528 (2022)
4. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: hyperplane-based Temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2001–2011 (2018)

5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
6. Gao, Y., Feng, L., Kan, Z., Han, Y., Qiao, L., Li, D.: Modeling precursors for temporal knowledge graph reasoning via auto-encoder structure. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, pp. 2044–2051 (2022)
7. Garcia-Duran, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4816–4821 (2018)
8. Goyal, P., Kamra, N., He, X., Liu, Y.: Dyngem: deep embedding method for dynamic graphs (2018)
9. Han, Z., Chen, P., Ma, Y., Tresp, V.: Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: International Conference on Learning Representations (2021)
10. Han, Z., Ding, Z., Ma, Y., Gu, Y., Tresp, V.: Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021)
11. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: International Conference on Learning Representations (2017)
12. Jin, W., et al.: Recurrent event network: global structure inference over temporal knowledge graph (2019)
13. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion Proceedings of the the Web Conference 2018, pp. 1771–1776 (2018)
14. Leetaru, K., Schrodt, P.A.: Gdelt: global data on events, location, and tone. ISA Annual Convention (2013)
15. Li, Z., et al.: Complex evolutional pattern learning for temporal knowledge graph reasoning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 290–296 (2022)
16. Li, Z., et al.: HiSMatch: historical structure matching based temporal knowledge graph reasoning. In: Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 7328–7338 (2022)
17. Li, Z., et al.: Search from history and reason for future: two-stage reasoning on temporal knowledge graphs. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (2021)
18. Li, Z., 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, pp. 408–417 (2021)
19. Liu, D., et al.: User-event graph embedding learning for context-aware recommendation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1051–1059 (2022)
20. Mahdisoltani, F., Biega, J.A., Suchanek, F.M.: Yago3: a knowledge base from multilingual Wikipedias. In: Conference on Innovative Data Systems Research (2015)
21. Mao, H., Schwarzkopf, M., Venkatakrishnan, S.B., Meng, Z., Alizadeh, M.: Learning scheduling algorithms for data processing clusters. In: Proceedings of the ACM Special Interest Group on Data Communication, pp. 270–288. Beijing China (2019)
22. Miao, S., Liu, M., Li, P.: Interpretable and generalizable graph learning via stochastic attention mechanism. In: Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 15524–15543 (2022)

23. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
24. Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B.: End-to-end structure-aware convolutional networks for knowledge base completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 3060–3067 (2019)
25. Sun, H., Zhong, J., Ma, Y., Han, Z., He, K.: TimeTraveler: reinforcement learning for temporal knowledge graph forecasting. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021)
26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of The 33rd International Conference on Machine Learning, pp. 2071–2080 (2016)
27. Wang, S., Cai, X., Zhang, Y., Yuan, X.: CRnet: modeling concurrent events over temporal knowledge graph. In: Sattler, U., et al. (eds.) ISWC 2022. LNCS, vol. 13489, pp. 516–533. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19433-7_30
28. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2014)
29. Zhang, M., Xia, Y., Liu, Q., Wu, S., Wang, L.: Learning latent relations for temporal knowledge graph reasoning. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, pp. 12617–12631 (2023)
30. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: a graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 1234–1241 (2020)
31. Zhu, C., Chen, M., Fan, C., Cheng, G.: Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4732–4740 (2021)