IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 August 2023, accepted 6 November 2023, date of publication 16 November 2023, date of current version 28 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333365

RESEARCH ARTICLE

Recurrent Event Networks Based on Subgraph and Attention Enhancement

HONGXI LIU, JIANA MENG, AND SHICHANG SUN
College of Computer Science and Engineering, Dalian Nationalities University, Dalian, Liaoning 116600, China
Corresponding author: Jiana Meng (mengjn@dlnu.edu.cn)

ABSTRACT Temporal knowledge graph (TKG) reasoning, as an essential research direction in natural language processing, focuses on capturing the dynamic changes in entities and relationships over time. However, the inference task of predicting potential future events faces significant challenges, as it must deal with uncertainty, complexity, and missing data. To this end, this study proposes a new TKG extrapolation model SubRE-NET, based on the Recurrent Event Network(RE-NET). The model performs reasoning by aggregating local and global information, and introduces a subgraph in the encoding stage to enhance the ability to capture local correlations and temporal features of events within a time window. At the same time, the attention mechanism is introduced to solve the problem in which the original model cannot distinguish the importance of nodes and relationships, and the subgraph is further enhanced by cropping technology. In the aggregation stage, an extended relational graph convolutional network(RGCN) was adopted to overcome the limitations of the original model, which cannot capture temporal information when locally and globally aggregated. The experimental results show that, compared with the baseline model RE-NET, our SubRE-NET model achieved significant performance improvements on three event-based datasets and two public knowledge graph datasets, with an average MRR performance improvement of 11.49%. Simultaneously, the average performance of the Hits@1 metric is improved by 12.38%.

INDEX TERMS Attention network, time knowledge graph, subgraph, temporal extrapolation.

I. INTRODUCTION

Knowledge Graph (KG) is a multi-relational heterogeneous Graph with different edges and entities. With the wide application of knowledge graphs in environmental analysis [1], recommender systems [2], and biomedicine [3], link prediction has attracted considerable attention as an essential task in KG. In existing knowledge graphs, the relationships between entities are typically modeled using factual triples (subject, predicate, object), for example, (Macron, visit, Mongolia). However, its incompleteness has become a challenge because it is difficult for knowledge graphs to contain all possible facts in a domain. To compensate for this deficiency, researchers have proposed a method that uses prior knowledge for reasoning [4]. Research on static knowledge graphs [5] has made remarkable progress in this field, but real-world facts are not always correct or static. Therefore, to capture the temporal dependencies between facts better, it is particularly critical to introduce temporal information to construct a knowledge graph with temporal relationships (TKG) [6]. In TKG, each fact is presented as (subject, predicate, object, timestamp), such as (Macron, visit, Mongolia, 2023). Temporal knowledge graph reasoning faces significant challenges owing to the need to consider time. For example, Figure 1 shows that Macron had engaged in diplomatic activities with other countries in the past. This suggests the possibility of future relations with these countries, as well as potential sanctions, opposition, or visits.

To address this challenge, previous studies have extended static knowledge graph embedding and scoring function methods to facilitate temporal representation learning [7]. These methods employ time-dependent scoring functions to assess the potential for missing facts, resulting in substantial performance improvements. However, during the learning process, they tend to overlook structural features [8], [9], [10]. Subsequently, numerous approaches have emerged, such as TeMP [11] and T-GAP [12], which leverage multiple time-step snapshots of evolving graphs and the message-passing mechanism of Graph Neural Networks to enable dynamic representation and inference in temporal

The associate editor coordinating the review of this manuscript and approving it for publication was Maurizio Tucci.

knowledge graphs. Although these methods are capable of integrating structural and temporal information, they are primarily tailored for historical data and value prediction, making them unsuitable for forecasting future events.

In TKG, the process of predicting new facts corresponding to future timestamps is categorized as extrapolation reasoning [13]. Although research in this field is still in its early stages, it holds significant importance in many real-world industries, such as finance and logistics. In recent years, several models have been proposed to address the extrapolation reasoning problem in temporal knowledge graphs, such as Know-Evolve [14] and DyRep [15]. These models leverage temporal point processes to model the evolutionary behavior of facts in continuous time domains, resulting in a significant improvement in the predictive performance of the models. However, without fundamental facts from previous events, these models struggle to predict events corresponding to future consecutive timestamps accurately [16]. Furthermore, owing to complexities, instabilities, sparsity of data, and multiple temporal dependencies, the prediction of concurrent events within the same time window encounters various challenges.

The RE-NET [13] model was recently proposed as a novel approach to address the extrapolation reasoning problem. RE-NET defines the joint probability distribution of all events in TKG in an autoregressive manner and models all concurrent events within a time window as a local graph to infer the graph structure over several future time steps. However, using all the events within a time window as local information may lead to unnecessary model perturbations and increased computational complexity. In addition, the traditional RGCN [17] method is often used for node and relation aggregation in static knowledge graphs, and it is difficult to effectively capture the temporal dependencies between facts in TKG. These deficiencies limit the performance of the model in graph representation learning.

To overcome these limitations and better handle long-term sequences in temporal knowledge graphs, we propose the SubRE-NET model. This model introduces subgraphs and attention mechanisms [18], enabling the precise selection of critical information within local windows and enhancing the model's ability to capture temporal information, thereby making message propagation more flexible. In addition, we strengthen the subgraph structure through subgraph cropping techniques and adopt an extended RGCN to capture the evolution process of entities and relations. The primary contributions of the study are as follows:

(1) We propose the SubRE-NET model, which effectively captures the temporal dependencies and structural relationships in temporal knowledge graphs by introducing subgraph sampling within time windows and reducing the computational complexity.

(2) The model incorporates attention mechanisms to adaptively learn the importance of nodes and relations, thus reducing the model's sensitivity to noise and sparse data.

(3) We employed subgraph cropping techniques to optimize the subgraph structure, enhancing the model's capability to handle large graphs.

(4) By introducing an extended RGCN, the model can perform aggregation and inference in dynamic environments.

To validate the reliability and effectiveness of our approach in TKG link prediction tasks, we conducted experiments on five public TKG datasets, thereby yielding superior results. These findings demonstrate that the SubRE-NET model exhibits exceptional performance and potential for TKG reasoning tasks.

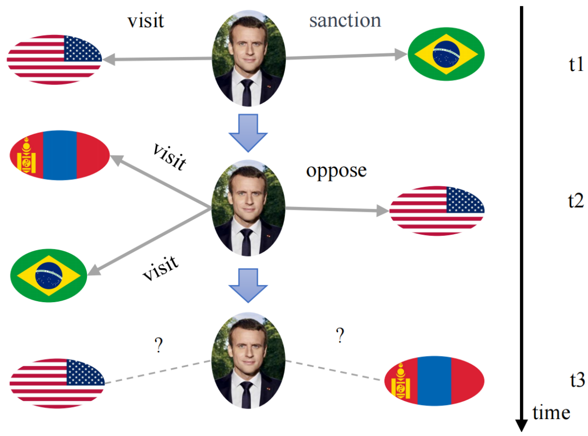

FIGURE 1. A subgraph of temporal knowledge about macron's international affairs.

Subsequently, this article is expanded according to the following structure: Section II presents related work to provide readers with a research background. Section III details the problems this paper aims to address. The fourth and fifth sections will analyze the RE-NET and SubRE-NET models in depth, respectively. Sections VI and VII detail the experimental design and analysis used to verify the model performance. Finally, Section VIII concludes the study and discusses future research directions.

II. RELATED WORK

A. STATIC KNOWLEDGE GRAPH REASONING

In terms of static knowledge graph reasoning, currently widely used methods mainly include embedding-based methods [19], [20] and neural network-based deep learning models [21], [22]. Embedding-based methods use scoring functions to evaluate the similarity between low-dimensional embeddings of relationships and entities. Simultaneously, neural network-based deep learning models such as GCN [23] and GraphSAGE [22] can automatically learn feature representations of entities and relationships, providing richer and deeper reasoning capabilities. Besides, there are other path-based [24] and rule-based reasoning methods [25]. However, these methods have shortcomings in directly modeling temporal knowledge graphs.

B. TEMPORAL KNOWLEDGE GRAPH REASONING

Reasoning with time knowledge graphs can be categorized into interpolation and extrapolation reasoning [13].

Interpolation reasoning [26], [27], [28] was used to predict and fill in the missing information in historical time intervals. Some works have attempted to add temporal information to static embedding modules for TKG inferences. For example, TTransE [9] extended the model of TransE [27] by considering the interaction of relations and time, and HyTE [10] associated timestamps with hyperplanes to explicitly integrate temporal information into the entity-relationship space. DE-SimplE [28] defines entity embedding as a function and provides a hidden representation of time. However, these methods are unsuitable for predicting events that correspond to future timestamps.

On the other hand, extrapolation reasoning aims at predicting future facts based on known prior information [29], [30]. For example, Know-Evolve [14] can learn entity representations that evolve nonlinearly in time order. DyRep [15] encodes temporal information into node embeddings through a dual-time-scale representation and a time-point process model. Although these methods have made some progress in temporal modeling, it is challenging to address the problem of modeling multiple concurrent events in the same time window. To solve this problem, GHNN [29] captures the changing sequence of graphs by extending the time points to model the probability of occurrence of events in continuous time. However, the GHNN is unsuitable for multistep reasoning since it mainly considers first-order subgraphs. In order to make up for this deficiency, RE-NET [13] modeled the corresponding events in the time window as a local graph, considered multi-hop structural information for time-series knowledge graph reasoning, and significantly improved efficiency. Based on the RE-NET model, The SubRE-NET model achieves more accurate modeling of time dependencies by introducing subgraphs and attention mechanisms within time windows while reducing the number of facts required to aggregate the time window data. The inferential capability of the temporal knowledge graph was further enhanced.

III. PROBLEM DESCRIPTION

Temporal knowledge graph reasoning plays a vital role in natural language processing tasks. This paper proposes a link prediction method suitable for inductive environments, aiming to solve the following issues:

Q1: How to efficiently handle the representation and aggregation of local information within a time window while maintaining time relevance and reducing model complexity?

Q2: The problem of distinguishing the importance of different nodes and relationships in constructing a subgraph.

Q3: How to reduce subgraph redundancy and prevent excessive model complexity.

Q4: How to dynamically aggregate the node and relation embeddings in the TKG to capture the evolution characteristics of time.

We conducted thorough experiments on various datasets to ensure the effectiveness of the model. To answer the above questions, we will answer them one by one in the SubRE-NET model.

IV. RE-NET MODULE

RE-NET is a TKG model for modeling temporal, multi-relational, and concurrent entity interactions. It infers information corresponding to multiple time steps in the future by defining the joint probabilities of all events. By computing the product of conditional probabilities for each event, we derive the joint probability distribution of event G corresponding to different timestamp t [13]. which includes the conditional probability $p(s|G_{t-m:t-1})_t$ of the subject the conditional probability $p(r|s, G_{t-m:t-1})_t$ of the relation, and the conditional probability $p(o|s, r, G_{t-m:t-1})_t$ of the object. as follows:

$$\begin{aligned} p(G) &= \prod_t \prod_{(s,r,o)_t \in G_t} p(s, r, o|G_{t-m:t-1})_t \\ &= \prod_t \prod_{(s,r,o)_t \in G_t} p(s|G_{t-m:t-1})_t \cdot p(r|s, G_{t-m:t-1})_t \\ &\quad \cdot p(o|s, r, G_{t-m:t-1})_t \end{aligned} \tag{1}$$

The RE-NET model employs two key components: an RGCN aggregator and recurrent event encoder. The RGCN learns low-dimensional embedding vectors of relations and entities and comprehensively considers global and local information to provide a rich contextual background for the input of the recurrent event encoder [17]. A recursive event encoder encodes a time series and can capture temporal dependencies between events [6]. Modeling the event sequence and learning time representation can help the model to understand the evolution process of the event and provide critical clues for predicting future information.

The global structure covers the complete information for all time steps, while the local structure only contains partial information within the time window. Local and global information complement each other, providing models with more accurate and refined prediction and inference capabilities. However, to obtain the local structure, all data corresponding to each time step within the time window needs to be used for feature extraction, which may face some challenges. First, doing so may lead to data redundancy, as multiple time steps within the same time window may contain partially duplicated information. Second, temporal relationships between entities may be lost. In addition, excessive computational complexity and imbalance of information corresponding to different timestamps within the time window may also be a challenge.

V. SubRE-NET MODULE

In this section, we provide a comprehensive introduction of the SubRE-NET model. First, we briefly introduce the construction of the structural diagram and the relationship between different components. Then, We describe each part of the model in detail. In addition, this section adds the question number after the title below in response to the four

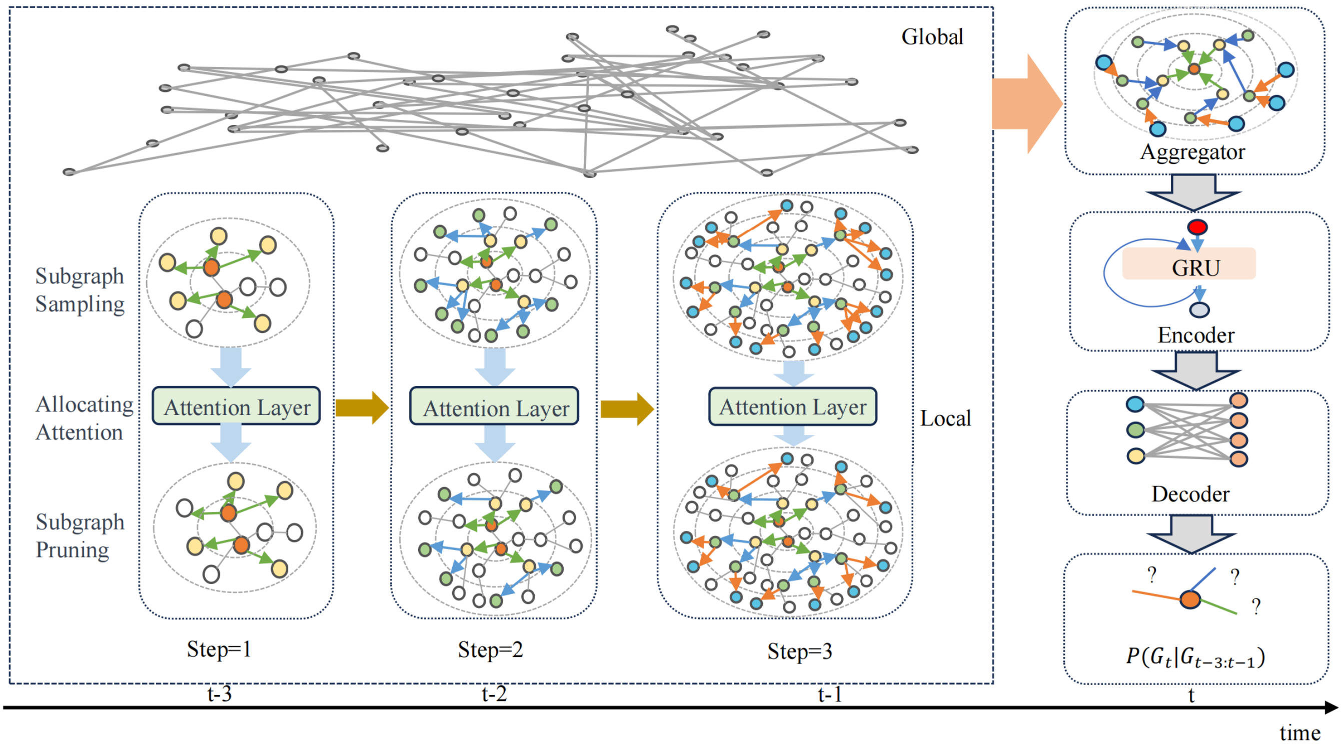

FIGURE 2. Here is the structural frame diagram for the SubRE-NET model. The "Step" value represents the number of iterations to create a subgraph in chronological order within a given time window.

questions that need to be addressed in the problem description section of this article.

A. GENERAL FRAMEWORK

SubRE-NET is a query-based extrapolation model that is used for link prediction. As illustrated in Figure 2, the model completes subgraph extraction in the time window through iterations, where each iteration includes three operations: subgraph sampling, attention allocation, and subgraph cropping. Specifically, following the time order, we sampled the neighboring nodes and relationships connected to the query target node from the period before the current timestamp. Different attention is assigned to the sampled nodes and relationships through the attention layer rather than to all the data of the entire time window. We clipped the sampled subgraph based on the attention score and repeated the process for the next iteration.

We obtained subgraphs with more comprehensive information by iterating three times within a given time window. The model regards the subgraph sampled in the time window as local features to capture the dynamic changes in nodes and relationships over time. At the same time, considering the static embedding vector as a global feature is convenient for charging the overall trend of the knowledge graph. Subsequently, the local and global information are aggregated using aggregation functions. Next, we use the GRU to encode the aggregated results, selectively updating and forgetting feature representations of nodes and relations through its gating unit. Completes the summary of the content of the previous time step and provides input information for the prediction results in the next time step. Finally, the decoder is responsible for completing the link prediction task.

B. SUBGRAPH EXTRACTION (Q1)

In the SubRE-NET model, the extraction of subgraphs can be regarded as the process of sampling events in a time window, in which the associated neighbor nodes and relationship types are selected according to the target nodes and time dependencies. We represent nodes as entity-time pairs, assuming that the target node to be queried is $v = (e_i, t)$, denotes the set of neighbor nodes before the current timestamp as N_v, and denotes the set of relation types as R_v. where $r_v \in \mathbb{R}_v$ represents a specific relation type before time step t. We propose three sampling strategies to obtain information on a fixed number of edges.

1) RANDOM NEIGHBOR SAMPLING

In the TKG, the random sampling strategy has the characteristics of fairness and balance. By performing random sampling, we can explore entities and relations equally over different periods to obtain unbiased information about entities and relations. The neighbors are sampled for each edge using a probability.

$$P(r_v) = \frac{1}{|R_v|} \tag{2}$$

here $|R_v|$ represents the total number of neighbor edges of the current node.

2) TEMPORAL WEIGHTED NEAREST NEIGHBOR SAMPLING

We adopted a time-sampling strategy, assigned different weights according to the time difference between neighbors, and smoothed them through an exponential function. Even neighbor nodes with small-time differences can obtain larger probability weights, thereby increasing the possibility of getting more information. This makes the steps closer to the current time more likely to be sampled. The sampling probability that we assign to each neighbor edge of the current node is

$$P(r_v) = e^{w(t'-t)} \tag{3}$$

where w represents the weight coefficient, and the neighbor timestamp t' is less than the time step t of the target node. When the number of relations is insufficient, we adopt the strategy of all sampling and ensure that there is no repeated sampling. In other words, all links related to the target node are selected as the only sampling results at the current time step.

3) FARTHEST NEIGHBOR SAMPLING

In TKG, the goal of the furthest neighbor sampling is to obtain a fixed amount of relationship information farthest from the target node, capturing relevant entities and relationships by only considering the most distant neighbors at the current time step. When the number of neighbor relations is less than the target sampling number, the sampling strategy of all and no repetitions is executed. The probability of each edge being sampled is:

$$P(r_v) = -\frac{1}{t - t' + \varepsilon} \tag{4}$$

here, ε is a small constant that prevents zero division in the denominator.

Compared to the above strategies, the second sampling method comprehensively considers the time difference between neighbor nodes, making the selection of neighbor relationships more flexible under different time granularities. In graph data, such a sampling method helps to extract time step information closer to the current time and obtain more meaningful features from neighbor information with a higher information value. Therefore, this is beneficial to our model. We conducted an ablation study in the experimental analysis section to compare three distinct sampling methods.

C. TEMPORAL ATTENTION LAYERS (Q2)

When extracting a subgraph related to a target node in a temporal knowledge graph, we face the challenge of distinguishing the significance of each node and its relationship to the target query node. However, we aimed to find evidence related to the target node. To avoid the problems of too large a sampled subgraph, high complexity, and information overload, the subgraph needs to be cropped. However, pruning is difficult due to the inability to accurately distinguish nodes and relations with equal importance. To address this issue, this study introduces an attention mechanism [18] in the subgraph, which takes the nodes and relations sampled from the subgraph as input and uses the attention mechanism to assign a query-based attention score to each link. In this manner, the essential information in the subgraph can be quickly screened out, resulting in the sampling of more valuable subgraphs. To comprehensively consider the temporal nature of a TKG, the relevance and similarity between different entities and correspondences are preserved. We used relation embeddings and node embeddings together as features to compute the edge attention scores, as follows:

$$\begin{aligned} e_{vu}^{l+1}(c, r_k) = (\mathbf{h}_v^l \cdot w_s^{l+1}) \odot (\mathbf{h}_{e_c}^l \cdot w_o^{l+1}) \odot (\mathbf{r}_k^l \cdot w_o^{l+1}) \\ \odot (\mathbf{r}_c^l \cdot w_o^{l+1}) \end{aligned} \tag{5}$$

where $e_{vu}^{l+1}(c, r_k)$ represents the attention score of edges (v, r_k, u) under the query $c = (e_c, r_c, ?, t_c)$, which is used to measure the importance of edges (v, r_k, u) in querying c. In addition, $\mathbf{h}_v^l$, $\mathbf{h}_{e_c}^l$, $\mathbf{r}_k^l$, and $\mathbf{r}_c^l$ respectively represent the hidden vectors of node v, query entity e_c, predicate r_k, and query predicate r_c in the hidden vector representation of the l^{th} inference step. Through the learnable weight matrices w_s and w_o, the model can flexibly fuse the query and node features to capture the correlation between them. For the hidden representation of the first layer, we use a random initialization method to assign initial values to nodes and relationships and then gradually learn better representations and relationships by optimizing these parameters, thereby improving model performance. We normalized the attention scores of the edges using the Softmax function.

$$\alpha_{vu}^{l+1}(c, r_k) = \frac{\exp(e_{vu}^{l+1}(c, r_k))}{\sum_{i \in \hat{N}_v} \sum_{r_j \in R_{vi}} \exp(e_{vi}^{l+1}(c, r_j))} \tag{6}$$

here, $\hat{N}_v$ represents the set of sampled neighbor nodes of node v, and R_{vi} refers to the collection of relationship types of all edges between node v and node i. as shown in Figure 3. We used the normalized attention score to perform weighted aggregation on the neighbor nodes of previous layer to obtain the current node's hidden representation.

$$\widetilde{\mathbf{h}}_v^{l+1}(c) = \sum_{u \in \hat{N}_v} \sum_{r_k \in R_{vu}} \alpha_{vu}^{l+1}(c, r_k) \mathbf{h}_u^l(c) \tag{7}$$

to fully consider the local characteristics of the node itself and the global relationship of the entire graph structure, we use the original vector representation of the node $\mathbf{h}_v^l(c)$ to fuse with the aggregated neighbor vector representation $\widetilde{\mathbf{h}}_v^{l+1}(c)$ and use the LeakyReLU activation function $\sigma(\cdot)$ Performed nonlinear transformation to obtain the vector representation $\mathbf{h}_v^{*l+1}(c)$ after feature fusion.

$$\mathbf{h}_v^{*l+1}(c) = \sigma(\gamma_1 \mathbf{h}_v^l(c) + \gamma_2 \widetilde{\mathbf{h}}_v^{l+1}(c)) \tag{8}$$

among them, The hyperparameters γ_1 and γ_2 adjust the contributions of the original node representation and the aggregated neighbor representation to update the current node representation, respectively, with the constraint $\gamma_1 + \gamma_2 = 1$. To maintain the semantic similarity of the vector

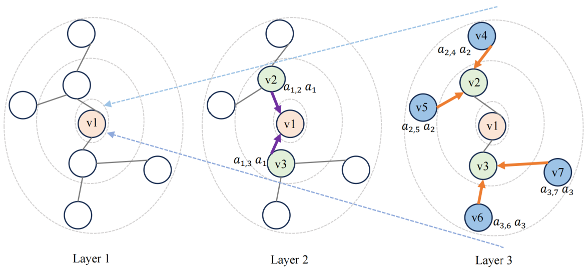

FIGURE 3. Update and aggregation of edges and nodes in subgraph sampling. The circles represent the level, and the arrows point to the direction of aggregation.

representations of relations and nodes while simplifying the model structure and decreasing the number of parameters, we applied the learnable parameter matrix W^{l+1} obtained during the node update process to update the neighbor relationship. The final node update representation $\mathbf{h}_v^{l+1}(c)$ is as follows:

$$\mathbf{h}_v^{l+1}(c) = W^{l+1}\mathbf{h}_v^{*l+1}(c) + b^{l+1} \tag{9}$$

b^{l+1} is the bias vector, and we obtained the updated relational representation $\mathbf{r}_k^{l+1}$ using the same linear transformation method.

$$\mathbf{r}_k^{l+1} = W^{l+1}\mathbf{r}_k^l + b^{l+1} \tag{10}$$

To realize the propagation of attention between edges and nodes, we use random initialization to assign an attention score to the target node and perform an inner product operation on the attention score of the current node and the attention score corresponding to the edge connected to it, to get the attention score corresponding to each neighbor node, to complete an attention propagation. After many iterations, each node received a propagated attention score. Ultimately, it completes attention allocation within the entire subgraph.

D. SUBGRAPH CROPPING (Q3)

After multiple iterations of sampling and attention allocation, we obtain a subgraph on a large scale. However, such large-scale subgraphs may lead to an increased consumption of model computing resources, extended training time, and likely exceed existing hardware limitations. To better adapt to the task requirements with limited computing resources, we use pruning technology to further optimize this subgraph. The pruning process removed redundant information and retained only the most critical information.

We iteratively clip the subgraph, select a fixed number of k edges with the most significant weight in the edge set R_{vu} obtained by sampling each time, and set the node characteristics corresponding to the unselected edges to 0 so that only the necessary node and edge information are preserved. As follows:

$$\widehat{R}_{vu} = \mathrm{topk}_a(R_{vu,k}) \tag{11}$$

where $\widehat{R}_{vu}$ represents a set of sampled edges. Then, in this manner, the operation is repeated for the nodes and edges after the next iteration has been sampled and attention allocated. We can end up with a more compact and efficient subgraph with iterative clipping. This helps the model better capture essential features and interrelationships in the graph structure.

E. NEIGHBOR AGGREGATOR (Q4)

In this section, we introduce two types of aggregation: mean pooling and time aggregation.

1) AVERAGE POOLING AGGREGATOR

The main idea is to average the characteristics of all neighbor nodes of entity v and use the obtained value to update the attributes of the current entity.

$$Agg(v) = \frac{1}{|\hat{N}_v|}\sum_{u\in\hat{N}_v}\mathbf{h}_u^l \tag{12}$$

Although average pooling aggregation is a simple and efficient way to aggregate neighbor nodes, it may ignore the complex interactions between nodes and the discrepancies between relationships when dealing with dynamic graphs.

2) TIME RGCN AGGREGATOR

Because the neighbor nodes of entities under each time step t may differ in TKG, timing information needs to be considered. To achieve this goal, temporal dynamic properties are incorporated into the RGCN [31], which is used to complete the dynamic aggregation of graph structural features in TKG. Specifically, we introduce the time step interval information $V_{s,o}$. For each time step t, we consider only the impact of events within the time interval on the entity to ensure that the events that occur between entities can follow the time sequence aggregate sequentially. Here is the formula:

$$\mathbf{h}_s^{(l+1,t)} = \sigma\left(\sum_{r\in R}\sum_{o\in {N_t}^{(s,r)}}\frac{1}{V_{s,o}}\cdot W_r^l\mathbf{h}_o^{(l,t)} + W_o^l\mathbf{h}_s^{(l,t)}\right) \tag{13}$$

the process of time aggregation can be understood as a two-layer convolutional operation. The inner layer aggregates the node feature, and the outer layer is used for relationship information mapping to complete the dynamic neighbor aggregation in the TKG. Where $\mathbf{h}_s^{(l+1,t)}$ represents the dynamic representation of the entity s after the time aggregation of the l^{th} layer at the t^{th} time step. ${N_t}^{(s,r)}$ represents the set of neighbor nodes o connected by entity s through relation r at time t. $V_{s,o}$ represents the time interval between entity o and neighbor node s. W_r^l is the weight matrix of layer l used to learn the influence of different relations r during neighbor aggregation. ${W_o}^l$ represents the self-loop weight matrix of the l^{th} layer, which is used to preserve the feature information of the entity itself. This aggregation method enables the model to better capture the historical evolution of nodes and the dynamic changes of relationships and obtain a more accurate time dynamic representation.To compare the impact of the aggregation method on the modeling, we did an ablation study in the experimental analysis section.

F. MODEL TRAINING

Our model integrates static and dynamic information from a temporal knowledge graph to enable effective inference of future events. SubRE-NET uses the GRU [6] model to encode the time series H^T obtained in the aggregation stage, aiming to obtain the hidden representation $\mathbf{h}^T$ of the last time step.

$$\mathbf{h}^T = GRU(H^T, \mathbf{h}^{T-1}, W) \tag{14}$$

here, $\mathbf{h}^{T-1}$ represents the hidden vector that corresponds to the previous time step. During the TKG inference process, in tackling the prediction of the subject, relation, and object, we treat each fact (s, r, o, t) as a multi-classification challenge. Each category represents existing relationships or nodes. Then, the model is normalized by utilizing the softmax function output and transformed it into a probability distribution, as shown below:

$$p_i = \mathrm{softmax}(\mathbf{h}^T) \tag{15}$$

p_i represents the probabilities of subjects, relationships, and objects. To measure the discrepancy between the forecast probability of the model and the actual distribution, we employed the cross-entropy loss function for parameter learning. This study adopted the same calculation method for the loss function as the RE-NET model [13]. The specific formula is as follows:

$$L = -\sum_{(s,r,o,t)\in G} \alpha \log p(s)_t + \beta \log p(r|s)_t + \log p(o|s, r)_t \tag{16}$$

among them, G represents the set of all facts, and α and β are super parameters, which are used to adjust the proportion of each loss term in the whole loss function. $p(o|s, r)_t$ represents the prediction probability distribution for each category of object o of the model when subject s, relation r, and time t are given. By minimizing this loss function, we trained the model to better predict the subjects, relationships, and objects of the unknown events.

VI. EXPERIMENT

This section introduces the data sets, evaluation indicators, baseline models, and related parameter configurations used in the experiment.

A. DATASET

To confirm the efficacy of our model, we assessed its performance on five publicly available datasets: WIKI [9], YAGO [32], ICEWS14 [14], ICEWS18 [33], and ICEWS05-15 [34]. Both WIKI and YAGO are knowledge graphs that contain time fact information, and the time granularity is one year. The WIKI dataset is based on Wikipedia, while the YAGO dataset is a cross-language knowledge graph. Both knowledge bases contain rich entities and relationship information. CEWS14, ICEWS18, and ICEWS05-15 are event-based datasets, all subsets of the ICEWS [35]. These datasets are organized on a daily basis. ICEWS14 and ICEWS18 correspond to events occurring in 2014 and 2018, respectively, while ICEWS05-15 covers global events from 2005 to 2015. These data sets offer comprehensive insights into international affairs and serve as valuable tools for researching early warnings and conflict analyses. We maintained the same method of data set partition as RE-NET. That is, we divided the data set into training, verification, and test sets according to chronological order. This ensured that the model did not come into contact with future information during the verification and testing phases, thus more accurately evaluating the performance of the model on unseen data. Table 1 shows more details about the dataset.

TABLE 1. Statistics of the datasets.

Dataset	N_{net}	N_{rel}	N_{train}	N_{valid}	N_{test}	Time interval
YAGO	10,623	10	161,540	19,523	20,026	1 year
WIKI	12,554	24	539,286	67,538	63,110	1 year
ICEWS14	7,128	230	63,685	13,823	13,222	1 day
ICEWS05-15	10,488	251	322,958	69,224	69,147	1 day
ICEWS18	23,033	256	373,018	45,995	49,545	1 day

B. EVALUATION INDEX

To quantify the performance of our proposed model, we chose two commonly used performance evaluation indicators: MRR and Hits@k. MRR is a comprehensive assessment criterion that considers the average reciprocal ranking of the model in the ranking task and the ranking of all correct answers.

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{\mathrm{rank}_j} \tag{17}$$

where Q is the total number of query samples and $rank_j$ is the ranking of correct answers in the j^{th} query sample. The values of MRR range from 0 to 1 and the higher the value, the better the model performs in sorting tasks. Hits@k more directly reflect whether the model contains the correct answer in the first k outputs. The values of k were generally 1, 3, and 10.

$$\mathrm{Hits@K} = \frac{1}{|Q|}\sum_{j=1}^{|Q|} p_k(j) \tag{18}$$

where $p_k(j)$ represents the probability of the j^{th} prediction hit, if the hit is 1, otherwise, it is 0. For the Hits@k indicator, the value range was also between 0 and 1, and the higher the value, the higher the accuracy of the model in the first k outputs. A comprehensive analysis of these indicators can provide essential information for a better understanding of the reasoning ability of the model. Furthermore, the filtering setting ignores time information and may mistakenly delete some valid facts [36]. To solve this problem, we used the time filtering setting [37], which only filters the actual triples that exist at the time of the query to improve the model's performance in practical applications.

TABLE 2. Here are the performance percentages for ICEWS14, ICEWS05-15, and ICESW18. The best results are highlighted in bold.

Datasets	ICEWS14				ICEWS05-15				ICEWS18			
Model	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
TTransE	3.11	17.32	34.55	13.43	5.00	19.72	38.02	15.71	1.92	8.56	21.89	8.31
TA-DistMult	17.09	30.22	45.41	26.47	14.58	27.92	44.21	24.31	8.61	18.41	33.59	16.75
DE-SimplE	24.43	35.69	49.11	32.67	25.91	38.99	52.75	35.02	11.53	21.86	34.80	19.30
TNTComplEx	23.35	36.03	49.13	32.12	19.52	30.80	42.89	27.54	13.28	24.04	36.91	21.23
RE-NET	28.68	41.43	54.52	38.28	31.26	46.85	63.47	42.97	19.05	32.44	**47.51**	28.81
CyGNet	23.69	36.31	50.67	32.73	25.67	39.09	52.94	34.97	15.90	28.28	42.61	24.93
TANGO-Tucker	17.30	29.07	44.18	26.25	-	-	-	-	19.35	32.17	47.04	28.68
TANGO-DistMult	16.36	27.26	41.35	24.70	-	-	-	-	17.92	30.08	44.09	26.75
SubRE-NET	**32.59**	**45.73**	**56.97**	**40.71**	**37.83**	**52.10**	**63.66**	**46.47**	**20.79**	**33.47**	46.33	**29.22**

TABLE 3. The performances percentage on YAGO and WIKI. The best results are highlighted in bold.

Datasets	YAGO				WIKI			
Model	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
TTransE	18.12	40.91	51.21	31.19	21.67	34.43	42.39	29.27
TA-DistMult	48.15	59.61	66.71	54.92	39.92	48.73	51.71	44.53
DE-SimplE	51.64	57.30	60.17	54.91	42.60	47.71	49.55	45.43
TNTComplEx	52.92	61.33	66.69	57.98	40.04	49.31	52.03	45.03
RE-NET	53.06	61.08	66.29	58.02	46.88	51.19	53.48	49.66
CyGNet	45.36	56.12	63.77	52.07	29.06	36.10	41.86	33.89
TANGO-Tucker	53.05	60.78	65.85	57.83	48.52	51.47	53.58	50.43
TANGO-DistMult	59.18	60.31	67.90	62.70	49.66	52.16	53.35	51.15
SubRE-NET	**80.85**	**88.57**	**90.32**	**85.45**	**68.67**	**77.80**	**79.61**	**73.38**

C. BASELINE MODELS AND PARAMETER SETTINGS

Many previous works have shown that temporal knowledge graph models exceed static models regarding temporal reasoning ability. Consequently, we moved away from the static model and instead utilized the following seven models proposed in recent years as benchmarks.

TTransE [9]: This model incorporates temporal information into its framework and uses distance measures to describe the degree of association between entities and relationships.

TA-DistMult [38]: This model can effectively learn time-aware representations of relations using recurrent neural networks, but there are challenges when dealing with compound regards in TKG.

DE-SimplE [28]: regards entity embedding as a function that can provide different representations for entities and relations in other time slices.

CyGNet [39]: Integrates generation and replication mechanisms, can use the global graph structure to predict triples, and simultaneously makes future predictions based on historically repeated facts.

TNTComplEx [40]: Model for learning node embeddings that scale with the number of timestamps by decomposing tensors into temporal and static parts.

TANGO [36]: Using the transition layer of the graph to simulate the change in the state of the nodes and edges over time can effectively capture time dependence.

RE-NET: Predicting future occurrences by modeling events as probabilities.

Considering the limited experimental resources and similar performance of existing baseline models in the TKG inference task, we focused solely on testing the SubRE-NET model's performance and did not conduct repeated experiments on other baseline models. The SubRE-NET model was trained by the cross-entropy loss function in Python3.10 environment using Torch2.0.1+CUDA11.7 and Numpy1.24 and optimized by Adam optimizer, in which the learning rate was set to 0.001.

While training the SubRE-NET model, we adjusted the values of α and β according to the characteristics of the loss function to meet the requirements of the research task.

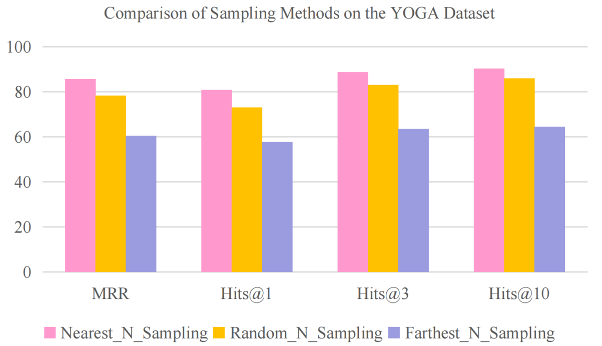

FIGURE 4. **Compare the impact of three different sampling methods on model performance on the YOGA dataset (in percentage). Where "N" stands for the abbreviation of Neighbor.**

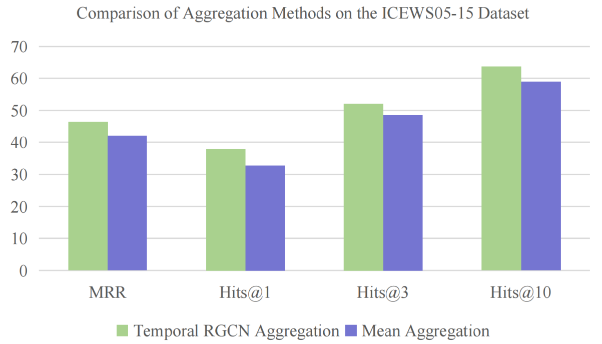

FIGURE 5. **Compare the modeling differences of different aggregation methods on the ICEWS05-15 dataset (in percentage).**

Expressly, we set α to 1 to highlight the model's focus on entity prediction and β to 0.1 to reduce the impact of relationships on the overall loss. We taught all the data sets for 20 iterations using a batch size of 128 and a time window size 10. This means that the model uses the past ten event sequences as inputs to predict future events. In the process of extracting the subgraph, we set the sampling step to 3. The corresponding nodes and relationships of each sampling are consistent in the time embedding dimensions, which are 128, 64, and 32, respectively, and during each iteration of selection, every node selects 12 edges. The sampling strategy uses time series weighted nearest neighbor sampling. The embedded dimensions of the global static nodes and relationships were 200. We used a two-layer RGCN aggregator for local and global aggregations. The GRU model was used as the cyclic event encoder in the coding phase. The number of GRU layers was one, and the hidden layer dimensions were 200. With the help of computing equipment equipped with a GeForce RTX3090 graphics card, we completed all the experiments.

VII. EXPERIMENTAL ANALYSIS

A. RESULT ANALYSIS

The effectiveness of the SubRE-NET model was verified by conducting an in-depth experimental analysis on multiple datasets. The experimental results show that, overall, the SubRE-NET model exhibits superior performance in prediction tasks, outperforming the baseline models. Compared to TTransE and TA-DistMult, DE-SimplE performs better time-dependent modeling owing to the introduced diachronic embedding. DE-SimplE and TNTComplEx are mainly suitable for filling in missing data in past time intervals, and there are some challenges in predicting future events, making them slightly inferior to the RE-NET model in terms of performance. The RE-NET model considers temporal relationships and fuses dynamic and static features for predictions. But it ignores how to extract dynamic information within the time window more efficiently. Our proposed SubRE-NET model outperformed RE-NET, TANGO-Tucker, CyGNet, and TANGO-DistMult in performance. One of the crucial reasons is the data sparsity in the data set, which shows the advantage of our model by introducing the attention mechanism in the time window.

Based on the presentation in Table 2, on the event-based dataset ICEWS14/05-15/18, especially in the ICEWS05-15 dataset, the SubRE-NET model significantly improved over the RE-NET model, and the MRR increased by 3.5%. Meanwhile, Hits@1 achieved a relative improvement of as high as 6.57%. In contrast, the progress in Hits@1 is more remarkable than that in Hits@10. Because in Hits@10, the model needs to make predictions on a broader range of possibilities, which is relatively more challenging, and it is more difficult to capture high confidence links. Table 3 summarizes the experimental results of the model on the public knowledge graph datasets WIKI and YAGO. Among them, compared to the RE-NET model, the MRR and Hits@1 indicators of the SubRE-NET model were improved by 25.57% and 24.79% on average, respectively. We observed a more significant performance gain for TKG reasoning on public knowledge graph datasets than on event-based datasets. This can be attributed to the more extensive and standardized nature of public knowledge graph datasets, which provide more abundant data resources for model training. In addition, these datasets also contain factual information that remains valid for a specific period, thereby further enhancing the predictive power of the model.

B. ABLATION EXPERIMENT

1) COMPARISON OF SAMPLING METHODS

We performed experiments on the YOGA dataset to analyze the effects of varying sampling methods on modeling. Figure 4 shows that the Temporal Weighted Nearest Neighbor Sampling performs the best, and the Farthest Neighbor Sampling performs the worst. This indicates that nodes closer in time can provide more valuable information for subgraphs, thereby affecting the modeling results.

2) COMPARISON OF AGGREGATION METHODS

We evaluated the influence of various aggregation techniques on the predictive performance of the model using the ICEWS05-15 dataset. Figure 5 summarizes the differences between the various aggregators in the other evaluation metrics. Since the extended RGCN fully considers the time

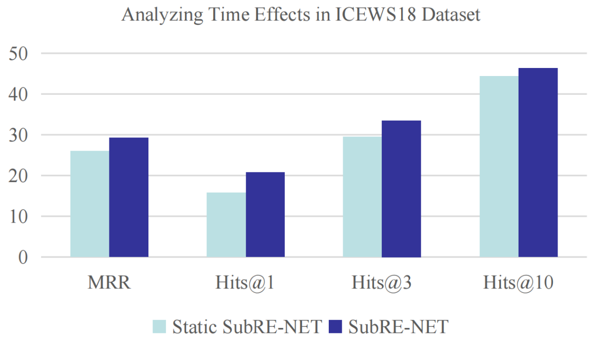

FIGURE 6. Exploring the varied impact of time factors on different metrics in ICEWS18 dataset (in percentage).

and the interaction between different complex relations, it outperforms the mean aggregation method. In addition, combined with Figure 4, we observed that using the extended RGCN aggregation method to aggregate neighboring nodes closer to the query entity during model training usually results in better prediction performance.

3) ANALYZING THE INFLUENCE OF TIME FACTORS ON MODEL BUILDING

In the SubRE-NET model, we compared the modeling results of retaining the time factor and removing the time factor in the ICEWS18 dataset to assess the influence of the time factor on the model's performance on the TKG link prediction task. Figure 6 shows that model performance dropped after the time factor was removed. This is because the model cannot extract local information within temporal windows in temporal order, and encoders lacking temporal information have limited capabilities. In addition, it can be observed that the values of the performance metrics of the two methods gradually approach each other over time. This trend was also reflected in the overall performance of the two methods.

VIII. CONCLUSION AND FUTURE WORK

This paper proposes a new TKG extrapolation model, SubRE-NET, which introduces a subgraph plus attention mechanism and an extended RGCN aggregator to enhance the link prediction accuracy in TKG. The subgraph plus attention mechanism helps the model concentrate on the sampling graph in the time window more accurately, which reduces the interference of redundant information and noise. The application of cutting technology enhances the ability of the model to deal with large graph structures. At the same time, the time-extended RGCN aggregator effectively captures the influence of time information on the evolution of relationships and improves the model's perception of time information. This study comprehensively evaluated the proposed model on five public datasets and verified its effectiveness through a large number of ablation experiments. The experimental results show that our model has made remarkable progress in relationship prediction and is better than the eight benchmark methods in performance.

In future work, we will further explore more improvement directions, expand the scope of application of the improved model, and apply it to more complex knowledge graph scenarios.

REFERENCES

[1] D. Gariglliotti, J. Bjerva, F. Å. Nielsen, A. Butzbach, I. Lyhne, L. Kørnøv, and K. Hose, "Do bridges dream of water pollutants? Towards dreamskg, a knowledge graph to make digital access for sustainable environmental assessment come true," in *Proc. WWW Companion, Companion ACM Web Conf.*, 2023, pp. 724–730.
[2] J. Liu and L. Duan, "A survey on knowledge graph-based recommender systems," in *Proc. IEEE 5th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Jul. 2021, pp. 2450–2453.
[3] J. Hao, C. J.-T. Ju, M. Chen, Y. Sun, C. Zaniolo, and W. Wang, "Bio-JOIE: Joint representation learning of biological knowledge bases," in *Proc. 11th ACM Int. Conf. Bioinf., Comput. Biol. Health Informat.*, 2020, pp. 1–10.
[4] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, May 2020.
[5] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, and F. Sun, "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multimodal," 2022, *arXiv:2212.05767*.
[6] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.
[7] Y. Ma, V. Tresp, and E. A. Daxberger, "Embedding models for episodic knowledge graphs," *J. Web Semantics*, vol. 59, Dec. 2019, Art. no. 100490.
[8] C. Xu, M. Nayyeri, F. Alkhoury, H. S. Yazdi, and J. Lehmann, "Temporal knowledge graph embedding model based on additive time series decomposition," 2019, *arXiv:1911.07893*.
[9] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *Proc. Companion Web Conf.*, 2018, pp. 1771–1776.
[10] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "HyTE: Hyperplane-based temporally aware knowledge graph embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2001–2011.
[11] J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton, "TeMP: Temporal message passing for temporal knowledge graph completion," 2020, *arXiv:2010.03526*.
[12] J. Jung, J. Jung, and U. Kang, "T-GAP: Learning to walk across time for temporal knowledge graph completion," 2020, *arXiv:2012.10595*.
[13] W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inference over temporal knowledge graphs," 2019, *arXiv:1904.05530*.
[14] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3462–3471.
[15] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "DyRep: Learning representations over dynamic graphs," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[16] C. Esteban, V. Tresp, Y. Yang, S. Baier, and D. Krompaß, "Predicting the co-evolution of event and knowledge graphs," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 98–105.
[17] W. Zhang, J. Chen, J. Li, Z. Xu, J. Z. Pan, and H. Chen, "Knowledge graph reasoning with logics and embeddings: Survey and perspective," 2022, *arXiv:2202.07412*.
[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[19] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.
[20] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 3060–3067.
[21] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, and Q. He, "Relational graph neural network with hierarchical attention for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 5, pp. 9612–9619.
[22] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[23] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Semantic Web 15th Int. Conf. (ESWC)*, Heraklion, Crete, Greece, Cham, Switzerland: Springer, 2018, pp. 593–607.
[24] R. Li and X. Cheng, "DIVINE: A generative adversarial imitation learning framework for knowledge graph reasoning," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 2642–2651.
[25] Y. He, Z. Wang, P. Zhang, Z. Tu, and Z. Ren, "VN network: Embedding newly emerging entities with virtual neighbors," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 505–514.
[26] S. Liao, S. Liang, Z. Meng, and Q. Zhang, "Learning dynamic embeddings for temporal knowledge graphs," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, Mar. 2021, pp. 535–543.
[27] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.
[28] R. Goel, S. M. Kazemi, M. Brubaker, and P. Poupart, "Diachronic embedding for temporal knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 3988–3995.
[29] Z. Han, Y. Ma, Y. Wang, S. Günnemann, and V. Tresp, "Graph Hawkes neural network for forecasting on temporal knowledge graphs," 2020, *arXiv:2003.13432*.
[30] Z. Li, S. Guan, X. Jin, W. Peng, Y. Lyu, Y. Zhu, L. Bai, W. Li, J. Guo, and X. Cheng, "Complex evolutional pattern learning for temporal knowledge graph reasoning," 2022, *arXiv:2203.07782*.
[31] Z. Li, X. Jin, S. Guan, W. Li, J. Guo, Y. Wang, and X. Cheng, "Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs," 2021, *arXiv:2106.00327*.
[32] F. Mahdisoltani, J. Biega, and F. Suchanek, "YAGO3: A knowledge base from multilingual Wikipedias," in *Proc. 7th Biennial Conf. Innov. Data Syst. Res.*, 2014, pp. 1–11.
[33] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, and M. Ward, "ICEWS coded event data," *Harvard Dataverse*, vol. 12, 2015.
[34] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng, "Temporal knowledge graph reasoning based on evolutional representation learning," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2021, pp. 408–417.
[35] J. Lautenschlager, A. Ruvinsky, I. Warfield, and B. Kettler, "Group profiling automation for crime and terrorism (GPACT)," *Proc. Manuf.*, vol. 3, pp. 3933–3940, 2015.
[36] Z. Han, Z. Ding, Y. Ma, Y. Gu, and V. Tresp, "Temporal knowledge graph forecasting with neural ODE," 2021, *arXiv:2101.05151*.
[37] Z. Han, P. Chen, Y. Ma, and V. Tresp, "Explainable subgraph reasoning for forecasting on temporal knowledge graphs," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–24.
[38] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," 2018, *arXiv:1809.03202*.
[39] C. Zhu, M. Chen, C. Fan, G. Cheng, and Y. Zhang, "Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4732–4740.
[40] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," 2020, *arXiv:2004.04926*.

HONGXI LIU received the bachelor's degree in software engineering from the Zijin College, Nanjing University of Science and Technology, Nanjing, China, in 2020. He is currently pursuing the master's degree in electronic information with Dalian Nationalities University, Dalian.

His current research interests include machine learning and natural language processing.

JIANA MENG received the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 2011.

She is currently a Professor with Dalian Minzu University, Dalian. Her current research interests include machine learning, natural language processing, and data mining.

SHICHANG SUN received the Ph.D. degree in computer science from the Dalian University of Technology, Dalian, China, in 2016.

He is currently a Vice Professor with Dalian Nationalities University, Dalian. His current research interests include machine learning and natural language processing.