Computer Science and Information Systems 22(1):153–180 https://doi.org/10.2298/CSIS240820006L

TPBoxE: Temporal Knowledge Graph Completion based on Time Probability Box Embedding

Song Li[1], Qi Wang[1], Zheng Li[2], and Liping Zhang[1,⋆]

[1] Harbin University of Science and Technology,
150080 Harbin, China
lisongbeifen@hrbust.edu.cn
wangqi441329@163.com
[2] Heilongjiang University, China

Abstract. Temporal knowledge graph completion is a technique that uses existing knowledge to predict or infer the missing information in the temporal knowledge graph. It combines the technical features of knowledge graph completion and time series analysis to deal with entities and relationships that change over time. The existing temporal knowledge graph completion technology fails to make effective use of the special relationship between relations and time series information, and it is difficult to fully represent the complex relationships existing in the graph. In order to solve the above problems, the model based on time probability box embedding (TPBoxE) was proposed. Firstly, the entities and relationships in the temporal knowledge graph are represented in the vector space by box embedding, so as to complete the static part of the temporal knowledge graph. Secondly, the head and tail entities that exist at the same time in a given time period are selected, and the completed static parts are filtered according to the time information of the entities. Finally, the Bayesian classification method is used to fully mine the time features hidden in the relationship, and the completion results are obtained by combining the confidence scores of the static parts. The link prediction task test of the proposed model on YAGO11k, WIKIdata12k, ICEWS18 and GDELT datasets shows that the proposed model has better performance than the existing excellent models, which proves the effectiveness and advancement of TPBoxE.

Keywords: Temporal knowledge graph, Knowledge graph completion, Temporal information, Link prediction, Bayesian classification.

1. Introduction

In 2012, Google launched the Google Knowledge Graph, which began to apply knowledge graph to search engines, marking the widespread application of knowledge graph technology. A knowledge graph is a vast knowledge base that contains information about entities (such as people, places, organizations, etc.), their relationships, and their attributes, which are structured and organized together to form an integrated knowledge graph. At present, the more complete knowledge graphs include DBpedia [10], Freebase [22], Wikidata [30], and YGAO [13]. Temporal knowledge graph (TKG) is formed into a time-stamped quadruple. Compared with the traditional knowledge graph, the temporal knowledge graph can express richer information.

⋆ Corresponding author

Nowadays, knowledge graphs have been widely used in natural language processing, search engines, question answering systems, and recommendation systems [21,2,23,31]. Baidu baike is an example of an applied knowledge graph, where users can search for keywords to obtain rich information including entity definitions, attributes, relationships, and various related knowledge.

However, real-world data is often incomplete and insufficient, and there may be missing or wrong information, which will inevitably lead to missing entities or relationships in the TKG. The above problems will reduce the reliability and application value of the TKG, and bring trouble and limitations to downstream application tasks. Therefore, it is of great significance to improve the data quality and content of the TKG.

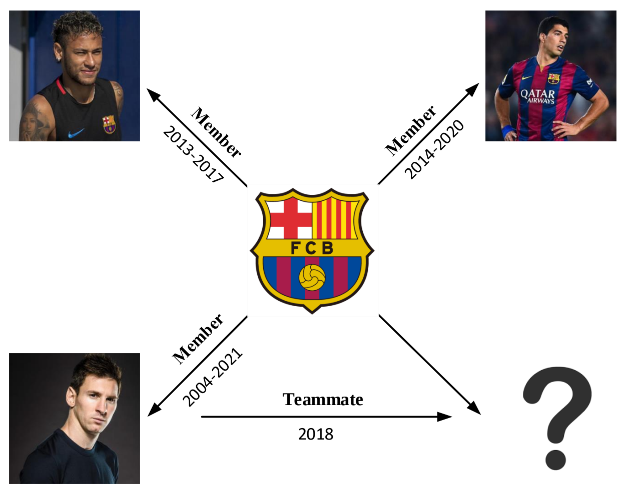

Fig. 1. An example of subgraph completion in a temporal knowledge graph

In order to improve the accuracy and completeness of the temporal knowledge graph, scholars have proposed various completion tasks. Temporal Knowledge Graph Completion (TKGC) refers to the filling or prediction of missing entities and relationships in an incomplete time series knowledge graph through inference, so as to make the TKG more complete and comprehensive. Compared with the traditional knowledge graph completion(KGC), the completion of the TKG needs to consider the consistency of time, that is, the existing TKG and the completed entity or relationship need to be logically consistent in time. In addition, the dynamic evolution of entities and relationships in the temporal knowledge graph is also very important. The completed entity or relationship needs to be able to satisfy the temporal correlation between the facts of the time series, including the order in which the facts occur, the time interval, and so on. As shown in Figure 1, the relationship between the connected entities has time information, indicating the point in time or time period when the fact is established. When the static knowledge graph completion method is adopted, it can lead to confusion in the semantic information of entities or relationships. For example, when completing a missing entity (Messi, teammate, ?, 2018), ignoring the given timestamp 2018 may confuse Luis Suárez with the rest of the Barcelona team, giving the wrong answer.

TKGC holds significant application value for time-related prediction, recommendation, inference tasks, and describing the development patterns of entities. Since the concept of TKGC was introduced, the technology for completing temporal knowledge graphs has rapidly advanced in response to practical application needs. However, most existing TKGC methods have one or more problems:

- Embedding the time series information into the entities or completely separating them, and ignoring the special connection between the relationship between the entities and the time series information, will lead to the loss of part of the information. These connections usually contain a wealth of knowledge, which can provide more accurate and reliable auxiliary information for reasoning, so as to improve the correctness of knowledge graph completion.
- The representation of time information is unreasonable, and only how to represent time information in the form of a point in time (e.g., [2022-12-25]) is not considered, but how to represent time information in the form of time period (e.g., [2022, 2023]) is not considered.
- The complex relationship between entities cannot be fully represented, such as: 1-n, n-n, etc, relationship.
- There are few studies on the relationship between entities and temporal information, and the importance of relationships is ignored in the process of completion.

In order to solve these problems and better improve the ability of KGC, this paper proposes a Temporal Knowledge Graph Completion based on Time Probability Box Embedding (TPBoxE). The model is based on the box embedding technology, which embeds the entities and relationships in the temporal knowledge graph into the vector space, and sets regular functions to capture the complex relationships between the entities. At the same time, a time scoring method is designed to filter out the correct static facts in a given time period according to the time series information of the entity. The model also has excellent universal applicability, and other static KGC models can be migrated to the TKGC task. The main contributions of this paper are briefly described as:

- In order to solve the problem of missing entities or relationships in the temporal knowledge graph, a temporal knowledge graph completion model TPBoxE based on time probability box was proposed, which improved the ability to learn the static part of the temporal knowledge graph through box embedding, and screened the static facts of the completion based on the time information as the evaluation basis, which solved the problem of applying the static knowledge graph completion method to the completion task of the TKG.
- The relationship between entities and temporal information is studied and verified by experiments. The feature-weighted frequency naïve Bayes classification method is used to integrate this feature into the screening process of static facts, which improves the accuracy of the completion results.

The proposed TPBoxE model is end-to-end in the TKG completion task, and the missing temporal facts can be automatically predicted by learning the existing TKG data as background knowledge, without the need to manually segment and label the data. For the missing temporal facts, different models output tasks are selected based on the type of missing data, such as entities, relations, or time information. The missing parts of the temporal fact quadruple are marked, and the final answer is found by following the scores of the TPBoxE model. In the completion process, the model will automatically learn the vector representations of entities and relations and their temporal information from the TKG background knowledge, and deduce the missing parts of the temporal knowledge through the above learned information. In this study, experiments are carried out on four public datasets extracted from the real time series knowledge graph WIKIdata, YAGO, ICEWS and GDELT, and the results in the link prediction task show that the performance of the TPBoxE model has obvious advantages over other models. It is compared with the existing excellent models, which shows that it has low time and space complexity. The superiority of the model in TKGC tasks was demonstrated.

2. Related Works

Knowledge graphs provide a structured way to organize and store data [25], correlate and represent a large amount of knowledge and information in the form of graphs, and provide a rich data foundation for artificial intelligence. However, there are still some problems with the completeness of the TKG itself, resulting in low availability. Therefore, scholars have carried out a lot of research on knowledge graph completion, which is divided into KGC and TKGC according to whether there is timing information inside the graph.

2.1. Knowledge Graph Completion

The model based on the translation approach embeds the entities in the KG into the vector space and encodes the relationships as translation work between the entities in the vector space. Representatives include TransE[4], and RotatE[26]. The classical TransE model embeds entities as vectors and represents relationships as displacements between entity vectors. This approach is easy to understand and implement but performs poorly when dealing with 1-n or n-n relationships. To better represent entities and relationships, RotatE embeds them in complex vector spaces and employs rotation operations to handle the complexity between relationships. This rotation operation allows it to better represent binary relations such as symmetry and self-inverse relations in the knowledge graph.

With the advances in artificial intelligence, many deep learning techniques are also being used to solve the KGC problem. ConvE [9] model is based on the convolutional neural network, which represents the KG as a two-dimensional matrix, and uses a multi-layer convolutional neural network to perform feature extraction on the matrix. By using convolutional layers and fully connected layers, ConvE can capture complex non-linear interactions between entities and relationships. DistMult [36] noticed the particularity of the relationship between entities, by representing the relationships as elements on the diagonal of the corresponding matrices, which makes it more advantageous in dealing with the task of KGC with symmetric relationships. Building on DistMult, ComplEx [27] uses more complex vectors to represent entities and uses complex inner products to predict the scores of triples, enabling it to represent and reason about more complex relationships. In recent years, with the wide application of attention mechanism [28], K-Bert [18] based on

Transformer architecture has been proposed to be applied to KGC tasks. It learns generic entity and relationship representations by large-scale pre-training on knowledge graph data with high quality. These representations capture the semantic information between entities and relationships well but cannot learn the structural information between entities and relations.

Knowledge graphs are a technical approach to describing knowledge and establishing relationships between everything in the world using graphical models [25]. The traditional KGC model cannot make good use of the graphic structure features in KG, which results in the loss of a large amount of information, including contextual information and entity features. For this reason, researchers have proposed some models to solve such problems, among which the representative ones are JGAN [14] based on graph convolutional neural networks, and GAT [29], which is a graph attention mechanism that applies the attention mechanism to graphs. The core of the JGAN and GAT model lies in its ability to learn the representation of a node in the graph by aggregating the features of the node and its neighboring nodes using a graph convolution model. This enables better use of information about nodes and edges in the graph structure data to learn the local features of entities. The Melo [34] model obtains meta-information and ontological information based on the contextual structure of entities, and infers high-confidence triples by mining logical rules within KG knowledge. This approach enables the model to understand how to represent KG appropriately.

2.2. Temporal Knowledge Graph Completion

Due to the inability of knowledge graphs to represent changes in entities over time, an increasing number of scholars are turning their attention to the task of TKGC. It is common practice to accomplish TKGC tasks by adding an embedded representation of temporal information [16] on top of existing KGC tasks.

Leblay et al. proposed the TTransE [17] model, which introduced the time dimension into the vector representation of entities and relations respectively based on TransE so that they could change dynamically according to the time interval. However, the shortcomings of TransE are also inherited, and the representation of entity, relationship, and time information is relatively isolated, resulting in some missing information. For this reason, Dasgupta et al. proposed HyTE [7], which is a hyperplane-based embedding method. The model gets the connection between entities, relationships, and temporal information by defining temporal information as a hyperplane and relationships as embedded as normal vectors in a hyperplane onto which entities are projected. Compared with TTransE, it has a better ability to deal with 1-n and n-n relationships but has some problems in the processing tasks of continuous dynamic graphs, large-scale knowledge graphs, and sparse data. TBoxE [20] proposed by Johannes Messner et al., based on the BoxE [1] model, embedded timing information into two transfer vectors to find the connection between entities again. By doing so, the missing relationships in the TKG can be predicted more accurately, and more complete temporal information can be provided. Existing TKGC methods embedding knowledge into Euclidean space often have the problem of high-dimensional nonlinear data and complex geometric structures. To solve this problem, Wang et al. proposed a multi-curvature adaptive embedding model MADE [33]. MADE models TKG with multiple geometries in multi-curvature spaces and assigns different weights to different curvature spaces through data-driven. In order to realize information interaction in multiple different geometric spaces, the model independently represents the embedding of entity, relationship and time information in a specific space, so as to fully capture semantic information and construct a quadruple distribution network to promote information aggregation and reasonable distribution between information.

The deep learning model has strong expressive ability and can learn more complex features and patterns of temporal knowledge graphs [5]. To capture the changes in the global characteristics of TKG, Mingsheng He et al. proposed a query-aware embedding model ConvTKG [11] based on convolutional neural network to execute TKGE, thus solving the TKGC task. In this model, a new temporal information encoder based on gated recurrent unit and attention mechanism is used to learn the query-aware representation of temporal information. For the positional semantic information of entities, the model assigns two independent vectors to each entity and makes use of inverse relations to allow them to be learned dependently.In order to deal with the problem of interdependence between timetable features and temporal facts, Yue et al. proposed the CEC-BD [37] model, which is based on tensor decomposition technology, which uses two-factor matrix and core tensor embedding to learn the entities and relationships in the temporal knowledge graph, and proposes a temporal smoothing function to represent the temporal information. The existing TKGC tasks still have the problems of ignoring the importance of temporal information and ignoring the location semantic information of entities, so Hao Wang et al. proposed a linear multi-view representation model MvTuckER [32]. The model treats various features in TKG as independent views and uses tensor operations to capture the relationship between different views, which greatly improves the model's ability to learn large knowledge graphs.

3. Proposed method: TPBoxE

The existing methods have not paid enough attention to the connection between the relationships and temporal information, resulting in insufficient utilization of knowledge and low accuracy of completion. Moreover, excellent KGC models cannot be directly applied to TKGC tasks. To improve the correctness of the complementation task and increase the generality of the method, this paper investigates the temporal characterization and embedding methods of entities and relations and proposes the TPBoxE model.

3.1. Basic Definition

Definition 1: TKG is usually stored as a quadruple [12], and for each fact of the quadruple, it is represented using $(h, r, t, [\tau_1, \tau_2])$. The head entity and tail entity are represented as h and t respectively, with r representing the relationship between entities, and τ_1, τ_2 indicating the start time and end time of the fact. where $h, t \in E, r \in R, \tau_1, \tau_2 \in T$. The temporal knowledge graph can be expressed as $G = \{(h, r, t, [\tau_1, \tau_2]) | h, t \in E, r \in R, \tau_1, \tau_2 \in T\}$. Fact G is valid only from the start time τ_1 to the end time τ_2, if and only if $\tau_1 = \tau_2$ means that the fact is valid only at τ_1 or τ_2 time-point.

Definition 2: TKG is typically represented as a collection of entities, relationships, and time points, denoted as E, R, and T respectively. Where $E = e_1, e_2, ..., e_n$ represents the set of entities, $R = r_1, r_2, ..., r_n$ represents the set of relationships between entities, and $T = \tau_1, \tau_2, ..., \tau_n$ represents the set of time points.

Definition 3: The head-time and tail-time of an entity in TKG denote the start time and end time of the entity's existence in the time dimension, respectively. For each entity, the set of time information can be represented as $T_e = \{\tau_1, \tau_2, ..., \tau_n | e, \tau_i \in G\}$, where head-time$= min(T_e)$, tail-time $= max(T_e)$.

3.2. The Overall Architecture of The TPBoxE Model

Firstly, the model represents knowledge by embedding entities as boxes and representing relationships as affine transformations between entity boxes. At the same time, the regular definition of relationships is carried out in this paper to enhance the learning of complex relationships, such as combinatorial, transitive and reflective relations. In addition, the negative sampling technique is used to optimize the training process of the model, further optimize the representation of entities and relationships, and obtain the answer set independent of time information. Secondly, a scoring approach for temporal information is proposed, which is mainly for evaluating how closely entities and temporal facts are linked in time. At the same time, we propose a method to categorize the connection between different relationships and temporal information. Filter the abnormal answers that contradict each other in time by the degree of dependence of the relationship on temporal information, keep compressing the size of the time-independent answer set, and ultimately obtain the set of answers to be complemented with the TKG. Finally, a score function for ranking quaternion confidence is designed to rank the set of answers.

The TPBoxE model proposed in this paper based on the assumption of translation invariance, models the entities in the TKG using high-dimensional vectors over the entire feature space for display modeling is represented as entity vector boxes, and relationships are represented as transfer vectors between entity vector boxes. In this process, the size and position of entity boxes are two important features of the entity representation process. The model determines the degree of correlation between the head and tail entities by the size and position of the head entity vector box after being updated by the relationship vector transfer and the size of the overlap interval of the tail entity vector box. At the global level, the goal of the TPBoxE model is to minimize the prediction error of all correct TKG static triples and to find the optimal entity and relation vector representations to be able to accurately reconstruct the TKG triples. The model scores of the triples and their temporal relevance are used as the main features, and the posterior probabilities of belonging to positive and negative samples in the static training samples are computed through a classification method. Finally, the samples are classified into the category with the highest a posterior probability. For the classification method proposed in this paper, the judgment on the classification of TKG complementary results can be continuously updated or adjusted by the training set data. This process is dynamic, and with the accumulation of data, the classification results continue to be accurate and reasonably inferred even in the case of scarce data.

The architecture of the TKG complementary model proposed in this paper is shown in Fig. 2 Subgraph (a) in the figure above represents the entities, relationships, and temporal information in the space of real temporal knowledge graphs. Subgraph (b) represents the correspondence of entities in the embedded vector space. Subgraph (c) represents the answer set without temporal information obtained by ignoring the time information and using the KGC method for the task "What buildings did China construct from year t_1 to year t_2?" for which the temporal knowledge is to be complemented. The temporal characteristics of the temporal facts in the TKGC task make it impossible to directly use the KGC method to complete the temporal facts. In TPBoxE model, the KGC method is used to obtain the static partial completion answer set of temporal facts in TKG. In order to select the correct answers that meet the constraints of time information from the answer set, TPBoxE learns the distribution probability of all entities in the TKG in the time dimension, and verifies each candidate temporal fact in the answer set according to the close relationship between the distribution probability and the relationship with the time information, and then obtains the time score of the temporal fact. The value range of the time score is between 0 and 1, and its size means the degree of matching of entities and relationships in the time dimension, and the higher the value, the greater the probability that the temporal fact is true. By setting a reasonable time scoring threshold by the proposed Bayesian classification method, the size of the answer set can be continuously compressed until the missing temporal fact completion result in the final TKG is obtained. In subgraph (d), the final answer is obtained by incorporating temporal information and continuously compressing the set size, which is represented using shaded areas.

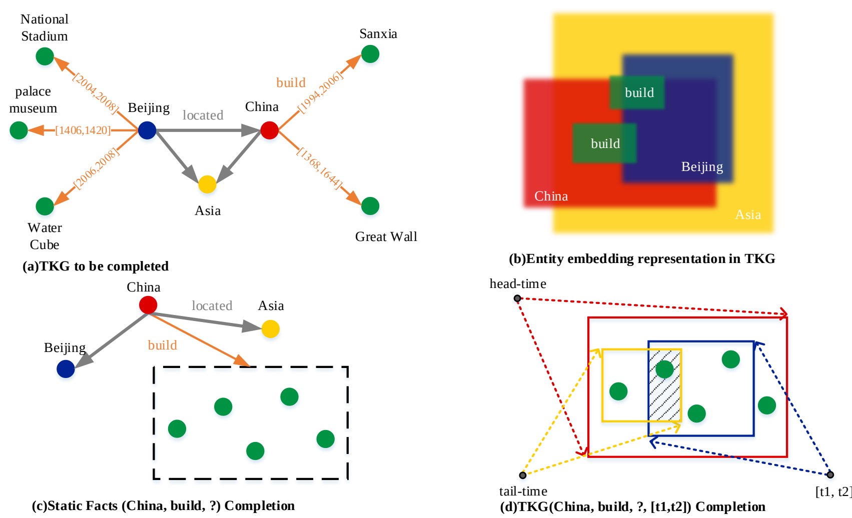

Fig. 2. Overall architecture of the TPBoxE Model

3.3. Knowledge Graph Embedding Method Based on Box Model

The traditional approach embeds entities and relations as vector dot products in a vector space, but the simplicity of the structure makes it limited in the relationships it can express. It lacks efficacy in dealing with relations expressing relationships with transitive properties such as $(A, r, B) \wedge (B, r, C) \implies (A, r, C)$, and the geometry of the box can make it better to model the relations between entities. Therefore, in this paper, we utilize the Box Embedding Model to make the entities in the TKG embedded as boxes and learn each relationship between entities as shape and position transformations between boxes. The formalization of a box embedding is defined as an n-dimensional hyperrectangle, which is the product between vectors:

$$Box(X) = \prod_{i=1}^{n} [x_i^{min}, x_i^{max}]. \tag{1}$$

Where $Box(X) \subseteq \mathbb{R}$, in some cases, two different boxes may have similar local structures, leading to the disappearance of their gradient signals, which in turn makes it difficult to train with gradient descent methods. In this paper, the Gumbel Boxes method [8,3] is used to train entity boxes, where the maximum and minimum coordinates of the box are denoted as:

$$x_i^{min} \sim MinGumbel(u_i^{min}, \beta), x_i^{max} \sim MaxGumbel(u_i^{max}, \beta). \tag{2}$$

In formula 2, u_i is a position parameter that determines the distribution location of the entity box. The exponential function in the Gumbel expectation formula is expanded using the first-order Taylor expansion as:

$$E(Box(X)) \approx \prod_{i=1}^{d} \beta log(2 + \frac{2len(Box(E))}{\beta} - 2\gamma). \tag{3}$$

In Formula 3, the $len(Box(E))$ parameter represents the length of the entity box E. The β represents the variance of the box and γ is the hyperparameter. In this research, by using $cen(Box(X)) \subseteq \mathbb{R}^d$ to denote the center point of the box, $len_h(Box(X)), len_t(Box(X)) \in \mathbb{R}^d$ to denote the offsets of the boxes of the head and tail entities, respectively. For the two cases where the same entity appears at the head or tail of the fact, we use the same box center, different box positions, and offsets to represent it. The resulting relationship between the position, center point, and offset of the entity boxes is:

$$\begin{aligned} u_i^{min} &= cen(Box(X) - len_h(Box(X))), \\ u_i^{max} &= cen(Box(X) + len_h(Box(X))), \\ u_j^{min} &= cen(Box(X) - len_t(Box(X))), \\ u_j^{max} &= cen(Box(X) + len_t(Box(X))). \end{aligned} \tag{4}$$

For head and tail entities in the same fact, $f_r(Box(X))$ is used as the mapping function of the relation, as follows:

$$f_r(Box(X)) = \begin{cases} len(Box(X)) \circ \alpha_r, \\ cen(Box(X)) \pm \Delta_e. \end{cases} \tag{5}$$

Considering that even if the same entity will have different semantic information at different positions, this paper adds the translation vector parameter Δ_e in the process of training the box representation vector, which is responsible for adapting to the semantic information transformation brought about by the transformation of the position of the entity. In order to avoid the problem that the entity representation in the TKG depends too much on the transformation of the relationship and cannot capture the nonlinear relationship, the model can learn a variety of semantic relationships of entities under different relations, and proposes the offset vector parameter α_r of the entity representation box according to the characteristics of the relationship, and enlarges or shrinks the size of the entity embedded in the box. The above parameters, together with the box vector embedded in the entity and the transfer vector of the relationship, constitute the representation of the temporal facts in TKG. The parameters α_r and Δ_e are initialized as all 1 and all 0 tensors, respectively, and the shape is consistent with the box vector and participates in the training with it. In summary, the scoring function formula for the model on entity or relation completion tasks is as follows:

$$score(h, r, t) = \frac{E[f_r(Box(h)) \bigcap Box(t)] * E[f_r(Box(t)) \bigcap Box(h)]}{E[Box(h)] * E[Box(t)]} \quad (6)$$

The score function simultaneously verifies the correctness of the static part completion results of temporal facts from both forward and inverse relation perspectives. The numerator represents the intersection of the head entity box and tail entity box in the binary random variable set after relation transition vector calculation. The denominator is the distribution set of binary random variables for the head entity box and tail entity box.

3.4. Regularized Constraints on Relationships

In the temporal knowledge graph, the relationship between entities is the core component, which determines the semantic information and structural relationship between entities. The representation of complex relationships has always been a major problem in the existing TKGC methods, and when the TKGC model often has the risk of overfitting in order to learn a reasonable relationship representation, the regular constraints of the relationship are helpful to simplify the model and improve the training efficiency of the model. In order to solve the above problems and ensure that the representation of complex relations in the temporal knowledge graph satisfies the specific semantic relations, and at the same time improve the robustness and generalization ability of the model, the regular constraints of relations are added to the training process of the model. The additional rules and constraints are imposed on special relationships, and relationship transfer between entities is achieved based on the similarity degree between them. Compared to previous work, the proposed method in this paper is more comprehensive and defines different rule constraints when treating multiple complex relationships. In addition, the method in this research is more robust and less affected by sparse and incomplete TKG data. Finally, it is possible to infer from the training dataset that the entities themselves have, but are not shown to express, implicit relationships in the TKG.

The work of Chen et al. [3] proves that the rule constraints of relationships are effective in improving the performance of the model, so this paper conducts further research on the basis of this. Using A, B, and C to represent entities, r, $r^{'}$, $r1$, $r2$, and $r3$ to represent relationships, and $Box(X)$ to represent the box representation learned by X entity.

Hierarchical Relations. The relation r holds if it holds for A, B, then the relation $r^{'}$ holds for A, B as well. The purpose of learning the relation r is that when (A, r, B) is true, then $(B, r^{'}, A)$ should also be predicted to be true. This objective is satisfied if for relations r and $r^{'}$ there exists entities A and B with the same probability that the facts they comprise hold. Therefore, in this paper, we constrain the relations r and $r^{'}$, with u denoting the set of embedded boxes, and the formula is expressed as follows:

$$L_e(r) = \frac{1}{|u|} \sum_{A,B \in u} \| P_{Box}(f_r(A)|f_r(B) - P_{Box}(f_{r'}(A)|f_{r'}(B))) \|^2 \tag{7}$$

Transfer Relations. The relation r can be passed between entities to each other, such that (A, r, B) holds if (A, r, C) and (B, r, C) hold, respectively. Then the following rule constraints are imposed on the boxes A, B, and C that imply the relation r:

$$L_t(r) = \frac{1}{|u|} \sum_{A,C \in u} \| P_{Box}(A|C)^{-1} \|^2 \tag{8}$$

Combinatorial Relations. The relations $r1$ and $r2$ can be deduced from $r3$, for $(A, r1, B)$ holds with $(B, r2, C)$, then $(A, r3, C)$ holds. To ensure that entities with combinatorial relationships remain unchanged after embedding into the vector space, we use the method of injecting rules to constrain the relationships as follows:

$$f_{r3} = f_{r1} \bigodot f_{r2} \tag{9}$$

Where $\bigodot$is the Hadamard product, i.e., the element-by-element multiplication. For any box $A \in u$ with a combinatorial relation, the expectation is that all have the following relation $f_{r3}(A) = f_{r1}(f_{r2}(A))$, which yields the rule constraint formula:

$$\begin{aligned} L_c(r_1, r_2, r_3) = \frac{1}{|u|} \sum_{A,B \in u} \| P_{Box}(f_{r3}(A)|f_{r1}(f_{r2}(B)))^{-1} \|^2 + \\ \| P_{Box}(f_{r1}(f_{r2}(B))|f_{r1}(A))^{-1} \|^2 \end{aligned} \tag{10}$$

In order to fully learn the background knowledge of TKG involved in learning and training, and to distinguish the correct and wrong temporal facts of the model, this paper uses the dynamic equilibrium negative sampling method for data augmentation in the process of model training. The main idea of this method is to make more negative samples of the same type in the next iteration of training to strengthen the learning of the weakness of the problem that it is difficult to distinguish specific samples in the process of each iteration of the model. Algorithm shows the specific process of the dynamic equilibrium negative sampling method:

```
Dynamic equilibrium negative sampling method (DEN_sample)
  Output The dataset that will participate in the
         training in the next iteration;
  var    Training set data S;
         Test set data T;
         Number of relationships n;
         Trained model BoxTE;
         The size of the dataset that each epoch
```

```
       participates in the training N;
begin
  // num is used to record the amount of training.
     data for each relationship
  num=Array.full(n, 1);
  for (h, r, t, [T1, T2]) in T:
      // Stores the inverse of the difference between
         the model score and the ideal score.
      num[r]=1/abs(1 - BoxTE((h, r, t, [T1, T2])))
      or BoxTE((h, r, t, [T1, T2]));
  // The reciprocal of the difference is normalized
     and the number of training samples is allocated
     according to the N-value.
  num=normalization(num)*N;
  // Initializes an empty collection to store the
     training set data.
  train_data = set();
  for i in num:
      // Generate i samples for each relationship.
      for j=0 to i:
          // Randomly destroy head-tailed entities to
             make new negative samples.
          set.add((h, r, random(t' in S && t' != t),
          [T1, T2]) or (random(h' in S && h' != h),
          r, t, [T1, T2]));
   return train_data;
end.
```

The code in Algorithm specifies the specific process of the dynamic equilibrium negative sampling algorithm, the 1 to 4 lines are to calculate the number of each type of relational data participating in the training, and the 5 to 9 lines are the negative sample making process by randomly destroying the head and tail entities according to the number of data participating in the training. For positive samples involved in the training process, positive sample loss L^{+} is obtained. The loss from the negative sample is L^{-}. Combined with the losses arising from the logical constraints above, the expression is:

$$L_{logic} = \omega_e \sum_{r \in R_e} L_e(r) + \omega_t \sum_{r \in R_t} L_t(r) + \omega_c \sum_{r \in R_c} L_c(r1 + r2 + r3) \tag{11}$$

Where $R_e, R_t, R_c \in R$, represent the set of hierarchical, transfer, and combinatorial relations, respectively, and the corresponding $\omega_e, \omega_t, \omega_c$ parameters are the regularization coefficients. The training loss function $L = L^{+} + L^{-} + L_{logic}$ for the PBoxE model is finally obtained.

3.5. A Tolerant Intersection over Union Time Information Evaluation Method

The $gIOU$ [24] and $aeIOU$ [15] are two commonly used intersection over union metrics for target detection tasks in the field of machine vision, which are also applicable to the time prediction evaluation task and have achieved good results. Time2BOX [6] proposed the $gaeIOU$ evaluation method on this basis, and the main idea is that when the size of the prediction interval is kept constant, the metric scores of the prediction intervals decrease as the gap to the actual intervals increases in the absence of intersections, and increase as the gap to the actual intervals increases in the presence of large intersections. Using I^{gold} to denote the actual time interval and I^{pred} to denote the predicted time interval, $gaeIOU$ can be expressed as:

$$gaeIOU(I^{gold}, I^{pred}) = \begin{cases} \dfrac{D(I^{gold} \bigcap I^{pred})}{D(I^{gold} \biguplus I^{pred})}, D(I^{gold} \bigcap I^{pred}) > 0 \\ \dfrac{D^{'}(I^{gold} \bigcap I^{pred})^{-1}}{D(I^{gold} \biguplus I^{pred})}, otherwise. \end{cases} \tag{12}$$

Where $D^{'}(I^{gold}, I^{pred}) = max(I^{gold}_{min}, I^{pred}_{min}) - min(I^{gold}_{max}, I^{pred}_{max}) + 1$ denotes the size of the gap between the predicted and the actual intervals. The $gaeIOU$ makes some improvements for the case where the gap between the predicted interval and the actual interval is very large, but it does not take into account issues such as the size of the gap between the actual and prediction interval widths. To solve this problem, this paper proposes the tolerant Intersection over Union ($tIOU$). The main idea is to reduce the influence of the size of the prediction interval range on the $tIOU$ value when the prediction interval intersects with the actual interval. The Intersection over Union (IoU) metric in existing machine vision is strict regarding the difference between the predicted image size and the target image size for accurate target recognition, which is not applicable in the TKGC task. The reason is that when the predicted time information and the actual time information do not intersect in the TKGC task, the confidence of the time of the completed time series fact depends not only on the difference between the time information and the real time series fact, but also on the difference between the predicted time interval and the actual time interval. In order to solve the above problems, we use the tolerance coefficient λ to dynamically adjust the size of the predicted interval in the tIOU. The magnitude of λ depends on the difference between the predicted time interval and the actual time interval, $\lambda = 1$ when the two are equal, and tends to $\lambda = 0$ when the difference between the two is large, so as to realize the constraint of the predicted size of the time interval. When the predicted time interval intersects with the actual time interval, the influence from the size of the predicted time interval should be reduced. In order to achieve this requirement, the union of the two types of time intervals in the gaeIOU formula is converted to the size of the actual time interval only. Through the above methods, the constraints on the predicted time interval are more tolerant when the predicted time interval is similar to the actual time interval, and the constraints are stricter when the two types of intervals are quite different, so as to realize the dynamic adjustment of time confidence. When the predicted interval does not intersect with the actual interval, the influence from the size of the predicted interval range is enlarged. Based on $gaeIOU$, it is generalized to $tIOU$, which is denoted by the formula:

$$tIOU(I^{gold}, I^{pred}) = \begin{cases} \dfrac{D(I^{gold} \bigcap I^{pred})}{I^{gold}}, D(I^{gold} \bigcap I^{pred}) > 0 \\ \lambda \dfrac{D^{'}(I^{gold}, I^{pred})^{-1}}{D(I^{gold} \bigcup I^{pred})}, otherwise. \end{cases} \tag{13}$$

Where the coefficient $\lambda = [abs(I^{pred}_{max} - I^{pred}_{min} - I^{gold}_{max} + I^{gold}_{min}) + 1]^{-1}$ represents the difference between the predicted and actual interval widths. For the predicted intervals of [2015,2018], [2020,2022], [2016,2023], [2016,2017], and [2024,2028], respectively, and the actual interval I^{gold} is [2018,2022], the results of the three-time scoring methods are shown in Table 1 below. It can be seen that the difference between the results of the three scoring methods is small when the time interval is small, while when treating the interval with the largest time [2016,2023], there is a large difference in the performance of the three scoring methods, in which the results of both the aeIOU and $gaeIOU$ methods are 0.857, whereas the result of the $tIOU$ method is 1. Therefore, the $tIOU$ method has a better performance when treating time intervals spanning a wide range of predictions.

Table 1. Comparison of time-scoring methods

prediction interval	aeIOU	gaeIOU	tIOU	intersection
[2015,2018]	0.125	0.125	0.167	T
[2020,2022]	0.8	0.8	0.667	T
[2016,2023]	0.857	0.857	1.0	T
[2016,2017]	0.142	0.071	0.014	F
[2024,2028]	0.1	0.05	0.025	F

3.6. Temporal Filters for Feature-weighted Frequency-based Naive Bayesian Classification Methods

Bayesian classification [35] is a statistical classification method based on Bayes' theorem, which can predict the labels of test set data (data to be predicted) according to the statistical distribution of training set samples. The Naive Bayesian classification is a special form of Bayesian classification. It assumes that the dependencies between relationships are "naive", i.e., that each attribute is independent of the other attributes given the conditions, which greatly reduces the complexity of the problem. The naive Bayesian classification method achieves good performance in many cases. It can be expressed as:

$$h_{nb}(x) = argmaxP(c)_{c \in y} \prod_{i=1}^{d} P(x_i|c). \tag{14}$$

Where d is the number of attributes, x_i is the value of x on the ith attribute, and c is a sample of a class.

To select the correct temporal facts from the static fact set, this paper proposes a feature-weighted frequency-based naive Bayesian classification method to compute the confidence score C between the facts to be tested and the temporal information. For the temporal fact G and the static fact S, whether G is correct or not is jointly influenced by the size of the intersection of the temporal scope of action of the entities in G and the temporal scope of the query about G, and the confidence scores of the static fact S. In addition, considering that different relationships between entities do not have the same level of sensitivity to time, the time scores are integrated according to the different entity relationships.

When a certain attribute value does not exist simultaneously with other attributes in the training set, the classifier will get a probability value of 0. Data in temporal knowledge graphs are often missing, which leads to the absence of certain attribute values in the training set alongside other attribute values. For example, the distribution probability interval of the time score for the temporal facts formed by the relation r in the training dataset is $[0.7, 1]$. When the time score $t_i \in [0, 0.7)$ of a temporal fact containing relation r in the test set needs to be predicted, the $P(t_i|c)$ probability part value in the classifier is 0. Even if the static part of the temporal fact is true, the classifier will still judge the temporal fact as an incorrect completion result, leading to reduced completion accuracy. To solve the above defect in the effect of one attribute value being completely erased by other attributes, we introduce the Laplacian correction method. Specifically, denote N as the number of features that affect the temporal fact G, N_i as the number of possible categories in a given feature, D as the size of the training set data volume, D_c as a given feature in the training set, x_i as the possible values in a given feature, and f_i as the frequency of occurrence of this value in D_c. The prior probability $P(x_i|c)$ and the conditional probability $P(x_i|c)$ are defined as equation (15):

$$P(x_i|c) = \frac{\sum_{i=1}^{D} f_i |D_c, x_i| + 1}{|D_c + N_i|}, P(x_i) = \frac{\sum_{i=1}^{D} |D_c, x_i| + 1}{|D| + N_i}. \tag{15}$$

Although the Naive Bayes classification method has achieved good results in some tasks, it assumes that all features are independent, which is rarely the case in reality. For this reason, this paper proposes a new feature weighting method, W_i, which is used to improve classification accuracy. The relationship between the confidence scores of the static triples, the temporal information features of entities, and relationships must be positively correlated, so in this research, we use the Chi-square statistics feature weighting method to calculate the weights W_i.

Specifically, the weight size is indicated by the difference between the actual frequency T_p and the predicted frequency F_p. The larger the difference represents that the feature has a greater impact on the classification result, and the larger the weight it occupies. The procedure for calculating the influence weights of the score sizes of the triples on the establishment of the temporal facts is shown below. In this paper, we use c_{tt}, c_{tf}, c_{ft}, and c_{ff} to represent the four cases of fact true and score within a certain interval, fact true and score not within a certain interval, fact false and score within a certain interval, fact false and score not within a certain interval, respectively. The prediction frequency F_p and the weight W_i can be defined in the following form:

$$\begin{aligned} F_p &= \frac{(c_{tt} + c_{tf})(c_{tt} + c_{ft}) - c_{ff}}{N_i} \\ W_i &= \sum_{G} \sum_{D_c} \frac{(T_p - F_p)^2}{F_p}. \end{aligned} \tag{16}$$

Also, the probability values are converted to exponential form to prevent the numerical lower bound from overflowing. Finally, the above equation (14) can be rewritten as:

$$h_{nb}(x) = argmax_{c \in y}[logP(x_i) + \sum_{i=1}^{N} W_i f_i logP(x_i|c)]. \tag{17}$$

For the treatment of timestamps, in this research, we use $T(e)$ to denote the temporal range of action of an entity, and for every fact G for which there exists an entity $e, T(e)$ will completely contain its temporal interval, as defined by Eq:

$$T(e_i) = \begin{cases} T(e_i)_{min} = \sum_{e_i \in G_j}^{i<m, j<n} min(T(e_i)_{min}, G_j^{\tau 1}) \\ T(e_i)_{max} = \sum_{e_i \in G_j}^{i<m, j<n} max(T(e_i)_{max}, G_j^{\tau 2}). \end{cases} \tag{18}$$

The final formula to derive the confidence score C is:

$$C(S, t_{pre}) = h_{nb}(tIOU(T(S), t_{pre}), r_i|True) + h_{nb}(score(S)|True). \tag{19}$$

Where S stands for the static triples and $T(S) = T(e_h) \bigcap T(e_t)$ denotes the time horizon of action of the static triples. Whether the completion of the temporal series facts is true depends on the size of the confidence score. The confidence formula is mainly composed of two parts, namely the probability value score based on the static fact and the probability value score based on the time series information. For the probability value of the static fact, the confidence score of the static part of the completed time series fact is obtained through the KGE model, and then the probability of the static fact being true under the confidence score is obtained by using the proposed Bayesian classification method (formula 17), which can bridge the performance gap of different KGE models. The probability value of the temporal series fact composed of the temporal information and relationship of the entity in the temporal dimension (formulas 13 and 18) is obtained by Bayesian classification method. The confidence score composed of the above two important parts can comprehensively consider whether the completion of the timing facts is correct or not. Finally, the algorithm gives the learning process of the TPBoxE model.

```
Training the TPBoxE model
  Output Confidence ranking of the embedding vectors e
         for all entities and relations and all to be
         complementary temporal quaternions;
  const  Tbatch = 2000;
  var    Training dataset S;
         Entity dataset E;
         Relation dataset R;
         Timestamps dataset T;
         Number of training iterations N;
         Batch size batch;
         Embedding dimension d;
         Number of negative samples n;
         Relation regularisation coefficients;
```

```
begin
 e = [xmin, xmax] in Gumbel(x, y) for each e in E;
 r = init_embedding() for each r in R;
 for i = 1 to N do
     for each batch S in Strain do
        data = DEN_sample(S, T, n, BoxTE, N);
        Tbatch =Union {data, (h,r,t)};
     Calculate the score function score(Tbatch);
     Update embeddings e, r, t;
     end for
 end for
 for S = (h, r, ?, T), e in E do
     // Calculating confidence model is added
        to the answer set.
     Sc = C(S, T);
     // The score of the quadruple prediction
        result and the confidence level of the
        model is added to the answer set.
     Map = Map intersection {S, Sc};
 end for
 Sort(Map);
end.
```

The algorithm is the whole model training process. Lines 1 and 2 are embedding initialization of entities and relations. Lines 3 to 14 are for representation training of the model using the static part of the temporal facts to get a fuller representation. The algorithm in lines 15 to 20 obtains the final set of supplementary results by replacing entities, filtering wrong answers, and categorizing them.

3.7. Hyperparameter Selection and Its Time and Space Complexity Analysis

In this research, the Stochastic Gradient Descent (SGD) method is used to train the TPBoxE model. The $batch$ value is 2048, the learning rate l_r is 0.0001, the embedding dimension d is 300, and the number of iterations N is 1000. 25 negative samples are constructed for each positive sample, and the GUMBEL-BETA of the training gumbel box is set to 1. The regularized constraint parameters are 0.1, 0, 0 for the YAGO11k dataset; 0.1, 0, 0 for the WIKIdata12k dataset; 0.1, 0.1, 0.1 for the complex ICEWS18 dataset and 0.3, 0.2, 0.2 for the GDELT dataset.

Time complexity and space complexity are important indicators to measure the performance of a model. Traditional models such as RotatE and TComplex introduce the concept of complex numbers to obtain a better representation of entities or relationships, but this greatly increases the computational effort and the complexity of the model. Other models, such as HyTE's representation of temporal information, are imperfect in some respects, and the complexity of the model increases steeply when the data is denser. Complex models usually have greater expressive and learning capabilities. The uniqueness of the TPBoxE model proposed in this paper is that it strikes a good balance between expressive power and model complexity, i.e., the model still has excellent expressive power without requiring too much computational resources.

We analyze and compare the space complexity and time complexity of the TPBoxE model and several existing state-of-the-art TKGC models. In terms of space complexity, the number of most model parameters is related to the number of embedding dimensions, entities, relations, and timestamps, except for the T-Temp model. In TPBoxE model, time information is used as a way to filter static triples, reducing the embedding of timestamps. In addition, the number of parameters required

Table 2. Comparison of time complexity and space complexity of the TPBoxE model with other TKGC models

Model	Time Complexity	Space Complexity
TTransE	O(Nd)	d(E+R+T)
HyTE	O(Nd)	d(E+R)d(T)
TeRo	O(Nd)	2d(E+R+T)
ChronoR	O(Nd)	3d(E+R)+3(rR+T)
TIME2BoX	O(Nd)	2d(E)+d(R+T)
T-Temp	O(Nd)	2d(E+R)+(2rdR+dT)+k(2E+4R)
BoxTE	O(Nd)	d(E+R)+2d(T)
TLogic	O(Nd)	2d(E+R+T)
TPBoxE	O(Nd)+O(1)	2d(E+R)+k(E+R+T)

by the model for the way of processing temporal information is k(E+R+T). In general, k is much smaller than the embedding dimension d, and it can be ignored. Therefore, the space complexity of the TPBoxE model is much smaller than other similar TKGC models. For time complexity, the time consumption of all similar models is positively correlated with the embedding dimension and the number of iterations, so the time complexity is O(Nd). The model additionally computes the temporal information of entities and relationships than the comparable models, which requires an additional time complexity of O(1) at the constant level, so that the time complexity of the TPBoxE model is equal to that of the comparable models.

In conclusion, the TPBoxE model outperforms or equalizes its counterparts both in terms of time consumption and space consumption, and it has excellent performance. It provides new ideas for solving the problems of large-scale knowledge graph complementation tasks and models that are too large to be deployed and require massive computational resources. For detailed information, please refer to Table 2.

4. Experimental Process and Result Analysis

To evaluate the effectiveness of the TPBoxE model on the task of temporal knowledge graph complementation, four commonly used datasets are used as experimental datasets in this paper: YAGO11k, WIKIdata12k, ICEWS18, and GDELT. Cropping out edges that contain individual entities, ensures better connectivity within the temporal knowledge graph. The WIKIdata12k dataset contains 40,000 triples and 12,554 entities, which is twice the size of the YAGO11k dataset. ICEWS is a global event database and conflict prediction system widely used in international relations research, of which the ICEWS18 dataset is an updated version of the ICEWS database in 2018. The ICEWS18 dataset contains data on a wide range of events, political conflicts, social unrest, and other relevant information about the globe, and the dataset has a large number of complex relationships within it. The GDELT dataset contains events occurring at more than 200 million geographic locations worldwide from 1979 to 2012, with events updated at 15-minute intervals. The statistical information for each dataset is shown in Table 3.

4.1. Evaluation Criteria

In this paper, Mean Rank, Hits@1, Hits@3, and Hits@10 are used as evaluation metrics. The Hits@1 metric represents the number of correct answers predicted by the model to rank first. The higher the Hits@1, the better the representation model performs in learning and representing the

Table 3. Statistical information on datasets

Data set	YAGO11k	WIKIdata12	ICEWS18	GDELT
Entity	10623	12554	23032	7691
Relation	9	23	255	240
Training set	16408	32497	373018	1734399
Validation set	2050	4062	45995	238765
Test set	2051	4062	49545	305241
Interval	year	year	day	15 minutes

relatedness between entities. Hits@n is how many of the n most likely outcomes are correct in total. Its value ranges from 0 to 1, and the closer it is to 1 the better the predictive ability of the model.

$$Hits@k = \frac{1}{|S|}\sum_{i=1}^{|S|}\begin{cases} rank_i, if(rank_i < k) \\ 0, else \end{cases}(k = 1, 3, 10). \tag{20}$$

Mean Rank is a widely used evaluation indicator, which indicates the average ranking of a model in predicting the correct associated entities for a query entity. The lower the Mean Rank value, the more it indicates that the model ranks higher in predicting the correct associated entities of the query entity, that is, the model's prediction accuracy is higher. The formula is:

$$MR = \frac{1}{|S|}\sum_{i=1}^{|S|} rank_i \tag{21}$$

4.2. Experimental Settings

In this paper, the model fills entities or relations into missing positions iteratively to obtain the value of the static triplet score, and finally adds time constraint information to obtain the confidence score. The confidence score is used as the basis for fact ranking to obtain the set of predicted results.

In the YAGO11k dataset, Fig. 3 is obtained from the trained model and scoring method as follows. The horizontal coordinates represent the mapped values of the different relationships, and the vertical coordinates represent the time expectations of the relationship under the fact that it holds and the weight it carries in the data, respectively. In Fig. 3, it can be concluded that different relationships rely on temporal information to different degrees, which means that some relationships treat time leniently. For example, for Relationship 0 and Relationship 6, which have the largest gap in time expectations, the expectations are 0.98 and 0.506, respectively, nearly double the difference.

To avoid that the above phenomenon is caused by the excessive weight in the dataset where a certain relation is located. Taking out the facts containing two kinds of relationships, and two probability values are calculated by using the model and the time scoring method. The first is the probability value that the triplet model scores in the interval [0.96,1] when the timing facts hold; The second is to calculate the probability value of its time information within the interval [0.96,1] through the time scoring method, as shown in Fig. 4.

For relation 7, it can be seen that for the temporal fact that implies this relation only holds if the time score is approximately 1, and the higher the time score, the higher the probability that the fact holds. It can therefore be argued that such relationships are more demanding in terms of temporal information. For relation 8, although the temporal facts containing this relation also follow the law that the higher the temporal score, the higher the probability of the fact being true, there is also a probability of about 0.412 that the temporal fact will be true for the scoring interval [0.96, 0.99]. The requirement for temporal information is not as stringent for Relation 8 as compared to Relation

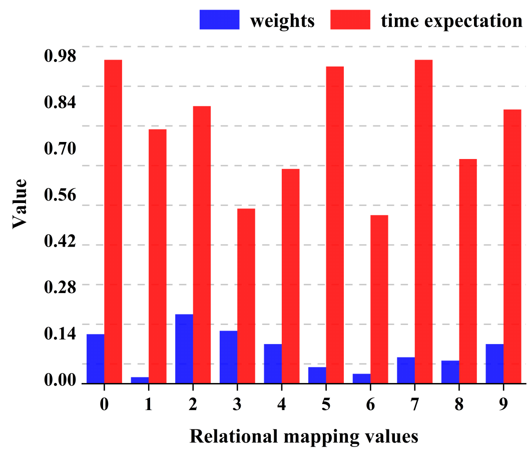

Fig. 3. The connection between relationships and temporal information in YAGO11k and their weights

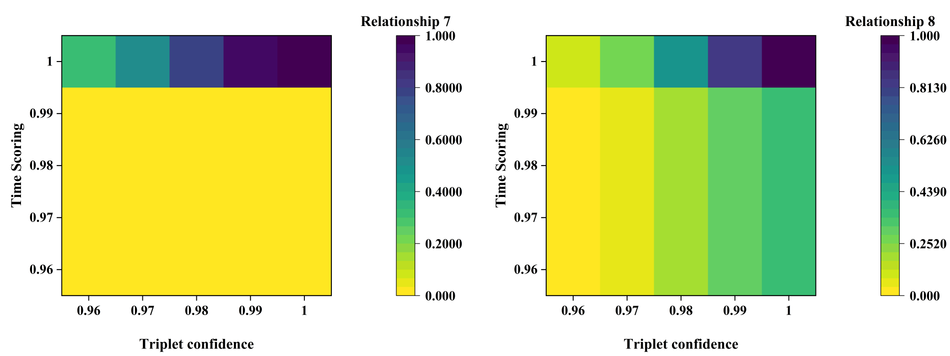

Fig. 4. Probability Plot of the effect of temporal information and static fact confidence on the establishment of temporal fact

7. We experimentally observe that the same KGC model for relations with more stringent temporal information requirements (e.g., ”relation 7”) lags behind that for relations with more relaxed temporal information requirements (e.g., ”relation 8”), which is one of the difficulties in applying the KGC model directly to the TKGC task.

4.3. Comparison Experiment

In the selection of benchmark models, considering the TPBoxE model as a variant of the box model, this paper takes BoxTE and TIME2Box as the benchmark models and compares them together with other TKGC models that perform well. The experimental procedure is to train TPBoxE on three datasets, and the optimal results are measured against the benchmark model according to three evaluation metrics: Mean Rank, Hits@1, Hits@3, or Hits@10.

Table 4. Results of the TPBoxE model on the head and tail entity link prediction task

Model	YAGO11k			WIKIdata12k			ICEWS18			GDELT			MR mean
	MR	H@1	H@10	MR	H@1	H@10	MR	H@1	H@10	MR	H@1	H@10	
TTransE	1233	.023	.149	1107	.057	.193	1461	.052	.213	1510	035	.168	1327
HyTE	1160	.026	.161	862	.035	.212	1003	.094	.292	1145	069	.182	1042
TeRo	824	.098	.287	539	.167	.423	82	.146	.355	1052	.097	.202	849.25
TNTComplex	318	.283	.517	361	.284	.496	482	.196	.317	821	.114	.219	495.5
TIME2BoX	232	.273	.561	198	.249	.511	-	-	-	-	-	-	-
BoxTE	201	.259	.497	236	.261	.501	347	.271	.395	569	.231	.372	363.25
TLogic	112	.287	.563	152	.292	.513	441	.201	.425	398	.213	.385	291
TKGC(SOTA)	112	.287	**.563**	106	.292	**.575**	347	.271	.425	186	.293	.485	234.75
TPBoxE	**86**	**.288**	.559	**77**	**.294**	.572	**166**	**.312**	**.511**	**144**	**.326**	**.487**	**118.25**

From the table 4 above, it can be observed that the TPBoxE model achieves state-of-the-art performance on the ICEWS18 dataset, and there exists a great advantage over other models. For the MR (mean rank) evaluation index, the model in this paper improves about 200 ranks compared with the optimal model BoxTE, and for the indexes Hits@1 and Hits@10, it also improves about 4 and 9 percentage points respectively, which is a big progress. The main reason for this is that the relationships in the ICEWS18 dataset are diverse and contain a large number of complex types of relationships, such as ”transfer” and ”self-reversal”. The other models involved in the comparison do not adequately represent complex relationships due to the simplicity of their modeling, and the TPBoxE model performs better because of the natural way in which boxes can model the particular connections between entities and relationships.

In the YAGO11k and WIKIdata12k datasets because of fewer and simpler inclusion relationships, the expressive power requirements of the models are not stringent, and therefore the gap between the TPBoxE model and the other models is not large. On the YAGO11k dataset, the MR and Hits@1 evaluation metrics achieve optimal performance, and the results under the Hits@10 evaluation metric have the same small gap with the performance of the current optimal model, which indicates that the model in this paper is competitive with the optimal model Tlogic[19].

In the WIKIdata12k dataset, the MR and Hits@1 metrics achieved optimal performance at the same time, and the Hits@10 metrics were almost identical in comparing the optimal performance.

For the GDELT dataset, which has the smallest time interval, the TPBoxE model achieves optimal performance and improves over the previous optimal results. Combining the model performances on all datasets, for the entity link prediction task, the models in this paper achieve optimal performance or are on par with the state-of-the-art.

Table 5. Results of the TPBoxE model on the relational link prediction task

Model	YAGO11k			WIKIdata12k			ICEWS18			GDELT			MR mean
	MR	H@1	H@3	MR	H@1	H@3	MR	H@1	H@3	MR	H@1	H@3	
TTransE	1.79	.775	.812	1.47	.821	.887	20.98	.233	.357	18.62	.102	.197	10.76
HyTE	1.68	.782	.836	1.42	.842	.898	20.13	.273	.329	16.91	.116	.202	10.03
TeRo	1.55	.771	.865	1.51	.841	.907	18.26	.302	.498	9.44	.269	.451	7.69
TNTComplex	1.41	.782	.862	1.39	.837	.865	17.56	.369	.502	11.64	.144	.233	8.00
TIME2BoX	1.32	.821	.924	1.22	.913	.922	-	-	-	-	-	-	-
BoxTE	1.44	.779	.839	1.41	.819	.857	16.19	.376	.513	9.62	.251	.366	7.16
TLogic	1.29	.839	.898	1.19	.901	.951	14.55	.349	.498	10.73	.207	.339	6.94
TKGC(SOTA)	1.24	.839	.924	1.19	.913	.951	14.23	.376	.513	9.62	.369	.451	6.87
TPBoxE	**1.21**	**.855**	**.967**	**1.09**	**.933**	**.991**	**11.21**	**.411**	**.579**	**7.42**	**.396**	**.457**	**5.23**

For the link prediction task of relational complementation, TPBoxE performs equally well. The Hits@10 evaluation metric was replaced with Hits@3 due to the presence of fewer relationships in the YAGO11k dataset and the WIKIdata12k dataset. For the YAGO11k dataset, this paper's model is optimal on all evaluation metrics, leading the baseline model by 0.03 on the MR metric, and by 2 and 4 percentage points for the Hits@1 and Hits3 evaluation metrics, respectively. On the WIKIdata12k dataset, the TPBoxE model is also at the optimum on all evaluation metrics, leading by about 10% on the MR metrics with a large advantage, and also leading by 2 points in Hits@1 and 4 points in Hits@3.

In the GDELT dataset, the lead is about 2 places on the MR metric, about 3 percentage points on the Hits@1 metric, and the Hits@3 metric is on par with the performance of the state-of-the-art model. Finally, on the ICEWS18 dataset, which has the most complex relationships, the model in this paper similarly leads and maintains a large advantage. It leads by about 3 places for the MR dataset, 4 percentage points for the Hits@1 metric, and 6 percentage points for Hits@3. The model in this paper achieves optimal results by performing the relational link prediction task on all the test datasets. For detailed information, please refer to Table 5.

The reason for the advanced results of the model is that the TPBoxE model has excellent portability of each component through structural modularization, and the excellent KGE model can always be used to achieve the best effect of the embedded representation of entities and relations in TKG. Based on the conclusion that the TKG completion results are a subset of the KG completion results, the model further improves the TKGC completion accuracy through the connection between entities, relationships and time series information. From the results of multiple experiments, it can be concluded that the greater the difference in time intervals, the more obvious the improvement of the TPBoxE model in the TKGC task compared with the existing KGC model and the baseline model.

4.4. Ablation Experiment

To analyze the impact of static knowledge graph complementation tasks and time-scoring methods on the model performance, this paper conducts several ablation experiments based on the model TPBoxE. First, in this paper, by replacing the static knowledge graph complementation method with other KG embedding methods, the resulting three model variants, are:

- TP-TransE: The box embedding method is replaced with TransE.
- TP-RotatE: The box embedding method is replaced with RotatE.
- TBoxE: The box embedding method is replaced with BoxE.

The above three static knowledge graph complementation methods are representative in terms of embedding method, and computational complexity. In addition, by replacing the ways of evaluation of temporal information into three ways such as gIOU, aeIOU, and gaeIOU, three variants of the model are obtained as TPBoxE (gIOU), TPBoxE (aeIOU) and TPBoxE (gaeIOU) respectively. All the above variants of the model are trained on the four datasets presented above. The variant model evaluates performance by linking prediction tasks through head and tail entities. The evaluation metric is MR^{-1}, with larger values indicating better modeling.

As can be seen in plot (a) in Fig. 5, the TPBoxE model has a large lead in the results under the MR^{-1} metrics on the four datasets. The TPBoxE model leads the state-of-the-art TBoxE model by about 30 percentage points on the YAGO11k dataset, and by 20, 10, and 20 percent on the WIKIdata12k, ICEWS18, and GDELT datasets, respectively, over the TBoxE model. On the KGC task, BoxE has a major advantage over other models in that the model expresses the complex relationships more adequately. Making it effective on the subsequent time-screening task. In Subfigure (b) of Fig. 5, for different temporal scoring methods, the tIOU method is 30 and 10 percentage points ahead of the optimal scoring method on the datasets YAGO11k and WIKIdata12k, respectively. While for the ICEWS18 dataset, the TPBoxE model is about 10 percentages behind compared to the optimal model. The reason for this is that the data within the dataset was taken from the 2018 Integrated Crisis Early Warning System (ICES) and the timeframe is from January 1, 2018, to December 31, 2018, which covers the entire 2018 timeframe, but there is a higher density of data within the same timeframe and a smaller time, so the difference in the results between the different scoring methods is smaller.

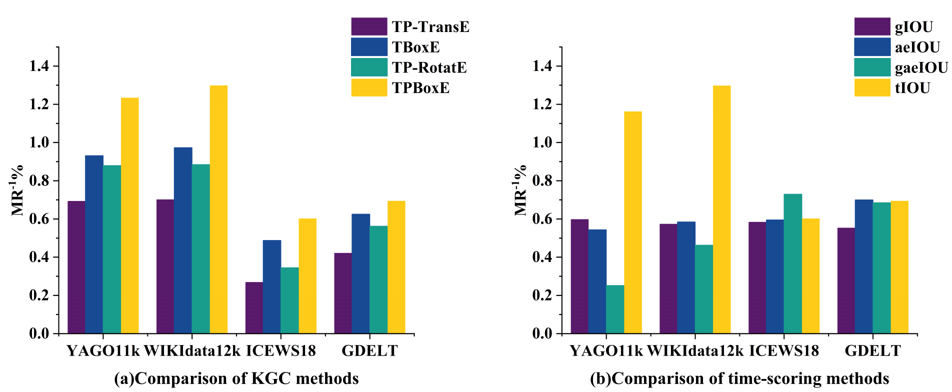

Fig. 5. Comparison chart of the results of ablation experiments

For the GDELT dataset, the fact that the start times within the dataset are all at the same moment similarly makes the difference in the results of the time scores smaller, resulting in the TPBoxE model only equaling the performance of the other similar models.

4.5. Parametric Analysis

In this section, this paper investigates the choice of parameters and their sensitivity of the TP-BoxE model, including batch size, learning rate, embedding dimension, number of negative samples, Gumbel-BETA for entity box embedding, and regularization constraint parameter for relations. The variation of TPBoxE's performance on the YAGO11k dataset is demonstrated by varying the hyperparameters.

Training batch size, learning rate, and embedding dimension. The training batch size and learning rate determine how fast the model is trained, and the embedding dimension determines the accuracy and expressiveness of the model. Subplot (a) in Fig. 6 demonstrates the effect of different sizes of parameters on the model. As larger embedding dimensions are used, the initial loss of the model is larger, and the training speed slows down at the same time as the learning rate and training batch decrease. The training of the model reaches the optimal loss and levels off when the parameters chosen are 2048, 0.0001, and 300, respectively.

Gumbel-BETA. The BETA parameter of the solid box is the scale parameter of the Gumbel distribution, which controls the width of the solid box distribution. In subplot (b) of Fig. 6, it can be seen that relative to 0.01, as the BETA parameter increases or decreases, it leads to the width of the bounding box distribution becoming too large or too small, thereby reducing the model's ability to complete missing information.

Regularization Constraints for Complex Relationships. For the complex relations in the dataset, this paper provides a special treatment for them, in which the regularization constraint parameter indicates the weights occupied by the complex relations. Take the YAGO11k dataset as an example, in which there are only hierarchical relationships and the number is 1, so the weight ratio between the three complex relationships is 1:0:0. For the relationship weight settings within the dataset, as shown in subfigure (c) of Fig. 6, the model achieves an optimal representation for entities and relationships within the YAGO11k dataset at a weight of 0.1:0:0.

Number of negative samples. Manufacturing negative samples as a method of data augmentation enhances the model's ability to differentiate between positive and negative samples. Subplot (d) of Fig. 6 demonstrates that as the number of negative samples corresponding to each positive sample increases, the correctness of the model increases. However, when the number of negative samples is after 20, the model accuracy improvement effect is smaller as the number of samples increases. Therefore, considering the model complexity, the number of negative samples in this paper is chosen as 25.

5. Conclusion

Along with the wide application of temporal knowledge graphs, its complementation task has also become one of the research hotspots. Because of this situation, this paper proposes a model based on the temporal probability box for complementing the temporal knowledge graph. The model proposed in this paper utilizes the full expressive power of box embeddings and accomplishes the task of temporal knowledge graph completion by introducing a Bayesian classification approach to model the linkages among static facts, temporal information, and time-series facts. Through extensive experiments on datasets from three different domains, this paper proves practically that the model has higher efficiency and better expressive ability in handling the complementary task compared to the traditional translation model, and theoretically demonstrates the existence of lower time and space consumption of the model.

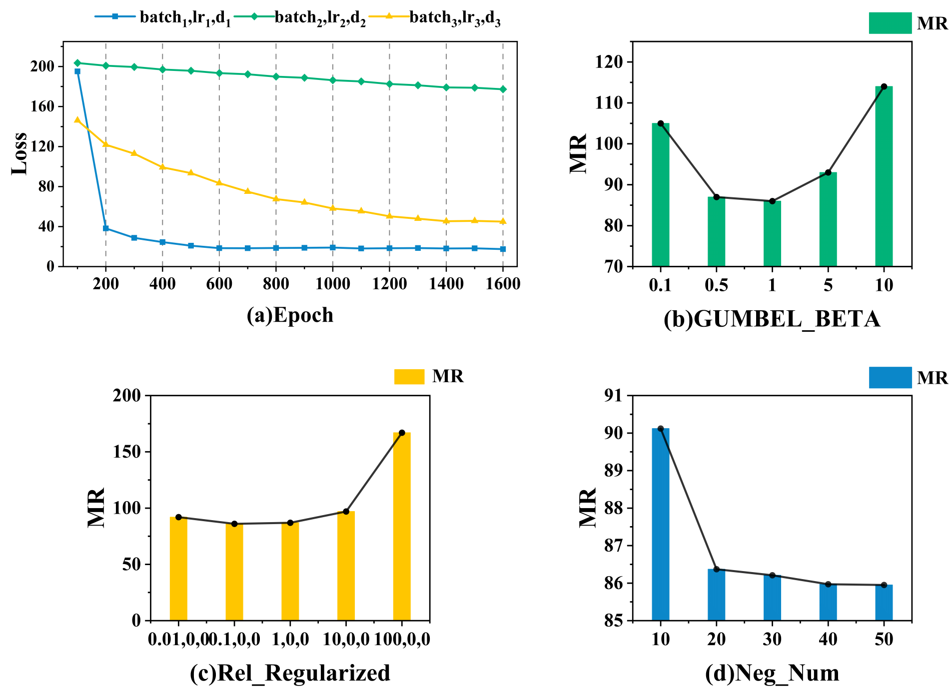

Fig. 6. Parameter sensitivity on TPBoxE

Most scholars have now noted the connection between entities and temporal information, but they have ignored the temporal properties of the relationship and the fact that it suffers from its problems of missing data and sparsity. In the future, we hope that we can further excavate the connection between entities, relationships, and temporal information to find better temporal representations and improve the expressive power of the model.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant No. 62072136, the Key research and development project of Heilongjiang Province No.JD2023SJ20, the National Key R&D Program of China under Grant No. 2020YFB1710200.

References

1. Abboud, R.: BoxE: A Box Embedding Model for Knowledge Base Completion. Advances in Neural Information Processing Systems 33, 9649–9661 (2020)
2. Ahmed, I.A., AL-Aswadi, F.N., Noaman, K.M., Alma'aitah, W.Z.: Arabic Knowledge Graph Construction: A close look in the present and into the future. Journal of King Saud University - Computer and Information Sciences 34(9), 6505–6523 (Oct 2022)
3. Boratko, M., Chen, M., Dasgupta, S.S., Li, X.L., McCallum, A.: Probabilistic Box Embeddings for Uncertain Knowledge Graph Reasoning. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 882–893. Association for Computational Linguistics, Online (2021)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data. Proceedings of the 26th International Conference on Neural Information Processing Systems pp. 2787–2795 (2013)

5. Cai, B., Xiang, Y., Gao, L., Zhang, H., Li, Y., Li, J.: Temporal Knowledge Graph Completion: A Survey. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. pp. 6545–6553. International Joint Conferences on Artificial Intelligence Organization, Macau, SAR China (Aug 2023)
6. Cai, L., Janowicz, K., Yan, B., Zhu, R., Mai, G.: Time in a Box: Advancing Knowledge Graph Completion with Temporal Scopes. In: Proceedings of the 11th on Knowledge Capture Conference. pp. 121–128. ACM, Virtual Event USA (Dec 2021)
7. 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. Association for Computational Linguistics, Brussels, Belgium (2018)
8. Dasgupta, S.S., Zhang, D., Li, X.L., Boratko, M., Vilnis, L., McCallum, A.: Improving Local Identifiability in Probabilistic Box Embeddings. advances in neural information processing systems pp. 182–192 (2020)
9. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D Knowledge Graph Embeddings. Proceedings of the AAAI Conference on Artificial Intelligence 32(1) (Apr 2018)
10. Färber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web 9(1), 77–129 (Nov 2017)
11. He, M., Zhu, L., Bai, L.: Convtkg: A query-aware convolutional neural network-based embedding model for temporal knowledge graph completion. Neurocomputing 588, 127680 (2024), https://www.sciencedirect.com/science/article/pii/S092523122400451X
12. He Peng, Gang, Z., Jing, C., Mengli, Z., Yuanlong, N.: Type-Enhanced Temporal Knowledge Graph Representation Learning Model. Journal of Computer Research and Development 60(4), 916–929 (2023)
13. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence 194, 28–61 (Jan 2013)
14. Huang, J., Lu, T., Zhu, J., Yu, W., Zhang, T.: Multi-relational knowledge graph completion method with local information fusion. Applied Intelligence 52(7), 7985–7994 (May 2022)
15. Jain, P., Rathi, S., Mausam, Chakrabarti, S.: Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 3733–3747. Association for Computational Linguistics, Online (2020)
16. Lacroix, T., Obozinski, G., Usunier, N.: TENSOR DECOMPOSITIONS FOR TEMPORAL KNOWLEDGE BASE COMPLETION. International Conference on Learning Representations pp. 1–12 (2020)
17. Leblay, J., Chekol, M.W.: Deriving Validity Time in Knowledge Graph. In: Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18. pp. 1771–1776. ACM Press, Lyon, France (2018)
18. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: K-BERT: Enabling Language Representation with Knowledge Graph. Proceedings of the AAAI Conference on Artificial Intelligence 34(03), 2901–2908 (Apr 2020)
19. Liu, Y., Ma, Y., Hildebrandt, M., Joblin, M., Tresp, V.: TLogic: Temporal Logical Rules for Explainable Link Forecasting on Temporal Knowledge Graphs. Proceedings of the AAAI Conference on Artificial Intelligence 36(4), 4120–4127 (Jun 2022)
20. Messner, J., Abboud, R., Ceylan, I.I.: Temporal Knowledge Graph Completion Using Box Embeddings. Proceedings of the AAAI Conference on Artificial Intelligence 36(7), 7779–7787 (Jun 2022)
21. Miao, L., Li, X., Yu, D., Ren, Y., Huang, Y., Cao, S.: Integrating users' long-term and short-term interests with knowledge graph to improve restaurant recommendation. Journal of King Saud University - Computer and Information Sciences 35(9), 101735 (Oct 2023)

22. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web 8(3), 489–508 (Dec 2016)
23. Peng, C., Xia, F., Naseriparsa, M., Osborne, F.: Knowledge Graphs: Opportunities and Challenges. Artificial Intelligence Review 56(11), 13071–13102 (Nov 2023)
24. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 658–666. IEEE, Long Beach, CA, USA (Jun 2019)
25. Song, L., Zhe, W., Liping, Z.: SL-tgStore: new temporal knowledge graph storage model. Journal of Zhejiang University (Engineering Science) 58(3), 449–458 (2024)
26. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: ROTATE: KNOWLEDGE GRAPH EMBEDDING BY RELA- TIONAL ROTATION IN COMPLEX SPACE. International Conference on Learning Representations pp. 1–18 (2019)
27. Trouillon, T., Welbl, J., Riedel, S.: Complex Embeddings for Simple Link Prediction. The 33rd International Conference on Machine Learning 48, 2071–2080 (2016)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All you Need. Advances in neural information processing systems 30, 11 (2017)
29. Velicˇkovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: GRAPH ATTENTION NETWORKS. stat 1050(20), 10–48550 (2018)
30. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Communications of the ACM 57(10), 78–85 (Sep 2014)
31. Wang, F., Gai, Y., Zhang, H.: Blockchain user digital identity big data and information security process protection based on network trust. Journal of King Saud University - Computer and Information Sciences 36(4), 102031 (Apr 2024)
32. Wang, H., Yang, J., Yang, L.T., Gao, Y., Ding, J., Zhou, X., Liu, H.: Mvtucker: Multi-view knowledge graphs representation learning based on tensor tucker model. Information Fusion 106, 102249 (2024), https://www.sciencedirect.com/science/article/pii/S1566253524000277
33. Wang, J., Wang, B., Gao, J., Pan, S., Liu, T., Yin, B., Gao, W.: Made: Multicurvature adaptive embedding for temporal knowledge graph completion. IEEE Transactions on Cybernetics pp. 1–14 (2024)
34. Wang, L., Lu, J., Sun, Y.: Knowledge graph representation learning model based on meta-information and logical rule enhancements. Journal of King Saud University - Computer and Information Sciences 35(4), 112–125 (Apr 2023)
35. Wenjun, Z., Liangxiao, J., Huan, Z., Long, C.: A Two-Layer Bayes Model: Random Forest Naive Bayes. Journal of Computer Research and Development 58(9), 2040–2051 (2021)
36. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: EMBEDDING ENTITIES AND RELATIONS FOR LEARN- ING AND INFERENCE IN KNOWLEDGE BASES. International Conference on Learning Representations pp. 1–13 (2015)
37. Yue, L., Ren, Y., Zeng, Y., Zhang, J., Zeng, K., Wan, J., Zhou, M.: Complex expressional characterizations learning based on block decomposition for temporal knowledge graph completion. Knowledge-Based Systems 290, 111591 (2024), https://www.sciencedirect.com/science/article/pii/S0950705124002260

Song Li received the Ph.D. degree in computer application technology from Harbin University of Science and Technology, Harbin, China, in 2009. He is currently a Professor with the College of Computer Science and Technology, Harbin University of Science and Technology. He is the author of two books, more than 60 articles. His research interests include spatial database, data mining, big data and information privacy protection, knowledge graph.

Qi Wang, a postgraduate student, is currently studying at the School of Computer Science and Technology, Harbin University of Science and Technology. His research interests include Temporal Knowledge Graph, Graph Neural Network, big data, and spatial database.

Zheng Li, a postgraduate student, is currently studying at the School of Information Management, Heilongjiang University. Her research interests include big data, data mining, and data analysis.

Liping Zhang received the M.S. degree in computer application technology from Harbin University of Science and Technology, Harbin, China, in 2006. She is currently a Professor with the College of Computer Science and Technology, Harbin University of Science and Technology. She is the author of two books, more than 40 articles. Her research interests include spatial database, data mining, knowledge graph, big data and information privacy protection.

Received: August 20, 2024; Accepted: November 20, 2024.