

## RotatS: temporal knowledge graph completion based on rotation and scaling in 3D space<sup>①</sup>

YU Yong (余泳)<sup>\*\*\*</sup>, CHEN Shudong<sup>②\*\*\*</sup>, TONG Da<sup>\*\*\*</sup>, QI Donglin<sup>\*\*\*</sup>, PENG Fei<sup>\*\*\*</sup>, ZHAO Hua<sup>\*\*\*</sup>

(<sup>\*</sup> Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100191, P. R. China)

(<sup>\*\*</sup> University of Chinese Academy of Sciences, Beijing 100191, P. R. China)

(<sup>\*\*\*</sup> Beijing Institute of Tracking and Telecommunications Technology, Beijing 100191, P. R. China)

### Abstract

As the research of knowledge graph (KG) is deepened and widely used, knowledge graph completion (KGC) has attracted more and more attentions from researchers, especially in scenarios of intelligent search, social networks and deep question and answer (Q&A). Current research mainly focuses on the completion of static knowledge graphs, and the temporal information in temporal knowledge graphs (TKGs) is ignored. However, the temporal information is definitely very helpful for the completion. Note that existing researches on temporal knowledge graph completion are difficult to process temporal information and to integrate entities, relations and time well. In this work, a rotation and scaling (RotatS) model is proposed, which learns rotation and scaling transformations from head entity embedding to tail entity embedding in 3D spaces to capture the information of time and relations in the temporal knowledge graph. The performance of the proposed RotatS model have been evaluated by comparison with several baselines under similar experimental conditions and space complexity on four typical knowl good graph completion datasets publicly available online. The study shows that RotatS can achieve good results in terms of prediction accuracy.

**Key words:** knowledge graph (KG), temporal knowledge graph (TKG), knowledge graph completion (KGC), rotation and scaling (RotatS)

## 0 Introduction

With the increasing application of knowledge graphs (KGs), a large number of knowledge graphs have been constructed. KG consists of entities, relationships, attributes, and attribute values. Generally speaking, KG can be represented by a set of triples. The representation of triples is  $(H, R, T)$ , where  $H$  (Head) and  $T$  (Tail) are a group of head entities and tail entities, respectively, and  $R$  (Relation) is a set of relationships between them. Existing KGs include YAGO<sup>[1]</sup>, Freebase<sup>[2]</sup>, Wikidata<sup>[3]</sup>, and WordNet<sup>[4]</sup>, etc., which are used for intelligent search, social networks, deep question answering, etc. However, many KGs usually suffer from incompleteness, and the purpose of knowledge graph completion (KGC) is to add new triples in KGs, such as obtaining a new triple (COVID-19, belongs to, Class B infectious diseases) from existing knowledge graphs related to contagious diseases. The task makes KG more complete. Therefore, KGC has drawn wide-

spread attention.

At present, many KGC models do not take temporal information into account. However, in many KGs, the occurrence of many events is related to time. This type of knowledge graph is called temporal knowledge graph (TKG). Typically, TKGs are represented by a quadruple  $(H, R, T, \Gamma)$ , where  $H$  and  $T$  are a set of head entities and tail entities respectively,  $R$  is a set of relationships between them, and  $\Gamma$  is a set of time when the event occurs. For instance, There are many answers to the question (Obama, visit, ?), if the time factor is not taken into account. But if the time factor is considered, the answer to this question is specific, as (Obama, visit, China) took place on November 15, 2009, and (Obama, visit, Thailand) took place on November 18, 2012. Ignoring temporal information can create significant uncertainty to KGC and negatively affect downstream tasks.

Traditional KGC models, such as TransE<sup>[5]</sup>, DistMult<sup>[6]</sup>, RotatE<sup>[7]</sup>, ComplEx<sup>[8]</sup>, etc., learn the low-dimensional embedding of entities and relations, and

<sup>①</sup> Supported by the National Natural Science Foundation of China (No. 6187022153).

<sup>②</sup> To whom correspondence should be addressed. E-mail: chenshudong@ime.ac.cn.

Received on Nov. 18, 2022

model the relationships between entities to perform KG completion. None of these models has a way to model temporal information, thereby ignoring the time when the fact occurs. To address this problem, several researchers have proposed related models, including TTransE<sup>[9]</sup>, HyTE<sup>[10]</sup>, TA-TransE<sup>[11]</sup>, ATiSE<sup>[12]</sup>, TeRo<sup>[13]</sup>, TempCaps<sup>[14]</sup>, TKGC<sup>[15]</sup>, HTKE<sup>[16]</sup>, etc. These models obtain the time factor by encoding temporal information and integrating this information with entities and relationships respectively, so that the event is time-dependent.

However, the above mentioned models are merely extensions of traditional KGC methods, and lack the ability to extract the semantic information of time. As an example, the TeRo model defines the temporal evolution of entity embedding as the rotation of the entity in two-dimensional space from the initial time to the current time<sup>[13]</sup>, and then uses the TransE method to obtain the relationship between the head entity and the tail entity after the rotation. Like other TKG completion methods, this model only considers the evolution of the entity itself in two-dimensional space with respect to time, and the semantic association between entities, relations and time is not better obtained.

This paper proposes a new method based on rotation and scaling (RotatS) for temporal knowledge graph completion. The model maps entities, relations, and time into three-dimensional space, and defines the evolution of head entities through a combination of time and relations to obtain tail entities.

Through the mapping of three-dimensional space, entities, relations and time can obtain richer semantic information. Furthermore, this paper defines the temporal and relational evolution of entity embedding as the rotation and scaling of head entity to the tail entity in the 3D space, and the model defines the relationship between head entity and tail entity in 3D space relative to relations and time, and obtains rich semantic association among entities, relations and time.

The advantages of the proposed model includes:

(1) It solves the problem that entity cannot be well combined with relations and time in traditional methods. A general approach to solving TKG completion is proposed in this paper. For events of either timestamps or time intervals, different rotation axes are used respectively, and the relationship between head entities and tail entities in different types of events are defined.

(2) The model considers the roles of time and relation in evolution at the same time. The rotation can help to make the direction between the head entity and the tail entity closer. Similarly, the scaling can help to make the size between the head entity and the tail enti-

ty closer. Through this method, entities, relations, and time are hopefully better associated.

(3) The performance of the proposed method is evaluated on four public datasets. Experimental results show that this paper's approach has certain advantages compared with several existing methods.

(4) The paper also conducts a comparison experiment between rotation and scaling in 3D space and 4D space. The experimental results show that the RotatS model performs better in these cases.

## 1 Related work

### 1.1 Static graph completion

Static KGC models can be divided into translation-based models, tensor decomposition-based models and neural network-based models.

#### 1.1.1 Translation based models

KGC is performed by learning low-dimensional embedding representations of entities and relations in the knowledge graph. These models are mainly based on TransE for transformations of spatial mappings. TransE<sup>[5]</sup> defines relations as the transformation from head entity to tail entity, but can only handle one-to-one relationships. For relations such as one-to-many, TransH<sup>[17]</sup> establishes a plane and projects entities and relationships onto the plane. TransR<sup>[18]</sup> uses the projection matrix of a particular relation to project entities into the relation space. sTransE<sup>[19]</sup> operates different transformations for head entity and tail entity, and projects head entity and tail entity into separate spaces. TransD<sup>[20]</sup> sparses the relational mapping matrix, therefore reducing the amount of parameters. TranSparse<sup>[21]</sup> solves the problem of unbalanced distribution of relations in triples. The relations that occur infrequently only need to train the matrix with low sparsity, and the relations that occur infrequently need to train the matrix with dense sparsity. TransM<sup>[22]</sup> focuses on the structure of the knowledge graph and constructs the model by pre-counting the weights of each relation in the training set. RotatE<sup>[7]</sup> introduces the complex domain on top of TransE, which maps entities and relations into a complex space, and defines each relation as a rotation from head entity to tail entity.

#### 1.1.2 Tensor decomposition based models

These models decompose the score of the triples into a tensor product. RESCAL<sup>[23]</sup> expresses the relations with a full-ranking matrix, calculating the product between entity embeddings and relation matrices. This model is simple and effective, but the parameters are increased. Relations in DistMult<sup>[6]</sup> use diagonal matrices to reduce the number of parameters but cannot

model the directionality of relationships. Complex<sup>[8]</sup> introduces the complex domain and considers that head entity embedding of the given entity and tail entity embedding are conjugate relationships. Among them, RESCAL and DistMult cannot obtain asymmetric relationships, for instance, triples  $(h, r, t)$  and  $(t, r, h)$  always have equal scores.

### 1.1.3 Neural network based models

Neural network-based models perform KGC variously: ProjE<sup>[24]</sup> model uses the feed-forward network to model the KG. ConvE<sup>[25]</sup> first splices the embedding of head entities and the embedding of relations, then performs a multi-layer convolution operation, and finally scores all candidate sets. ConvKB<sup>[26]</sup> modifies the form of the input data of ConvE, then obtains the feature layer by convolution, and finally splices and scores it. ConvR<sup>[27]</sup> performs the convolution operation by using relations directly as convolution kernels.

## 1.2 Temporal graph completion

Recently, some researchers have found that adding temporal information to the knowledge graph can further improve the performance of KGC. Some models are derived from TransE, such as TTransE<sup>[9]</sup>, embedding time information into the score function. TA-TransE and TA-DistMult<sup>[11]</sup> obtained the temporal evolution of relations by stitching the embedding of predicates and the temporal embedding using long short term memory (LSTM). HyTE<sup>[10]</sup> treats timestamps as hyperplanes and projects entity and relation embeddings onto the hyperplane to obtain new representations. This model is the expansion of TransH with respect to time by adding temporal information to head entity, relation and tail entity respectively. TempCaps<sup>[14]</sup> proposes a capsule network-based model to construct entity embeddings by dynamically routing the retrieved temporal relations and neighbor information. TKGC<sup>[15]</sup> models global trends and local fluctuations by mapping entities and relations in TKG to approximations of multivariate Gaussian processes (MGPs). HTKE<sup>[16]</sup> proposes a new hyperplane-based temporal-aware KG embedding model that incorporates temporal information into the entity-relation space to more effectively predict missing elements in KGs.

## 2 Methods

In this section, the TKG completion problem is defined and the method RotatS is introduced in detail. This method integrates entities, relationships, and temporal information to solve the TKG completion problem better.

TKG completion is also called link prediction task. The goal of this task is to predict hidden quaternions using observable quaternions<sup>[28]</sup>. This task can be divided into two parts: given a quaternion representation of the TKG as  $(h, r, *, \Gamma)$ , predicts the tail entity, and predicts the head entity given a quaternion representation of the TKG as  $(*, r, t, \Gamma)$ .

Quaternions have been used for static knowledge graph embeddings<sup>[29]</sup>. The definition and basic mathematical properties of quaternions are presented for a better introduction to the method proposed in this paper.

### 2.1 Quaternions

Quaternion<sup>[30]</sup> is the result of the expansion of the imaginary part of a complex number. Usually the quaternion  $\mathbf{q}$  is represented by  $(w, x, y, z)$ , where  $w$  is the real part;  $x, y$  and  $z$  are the imaginary parts, or it can also be expressed as  $\mathbf{q} = [w, v] = w + xi + yj + zk$  ( $i, j$ , and  $k$  are imaginary units). Some of the rules and operations for quaternions are as follows.

Basic rules and derivatives are

$$i^2 = j^2 = k^2 = ijk = -1 \quad (1)$$

$$ij = k; jk = i; ki = j \quad (2)$$

$$ji = -k; kj = -i; ik = -j \quad (3)$$

The inner product of quaternions: the inner product between  $\mathbf{q}_1 = w_1 + x_1i + y_1j + z_1k$  and  $\mathbf{q}_2 = w_2 + x_2i + y_2j + z_2k$  is the sum of the products of the corresponding real and imaginary parts.

$$\mathbf{q}_1 \cdot \mathbf{q}_2 = w_1 \cdot w_2 + x_1 \cdot x_2 + y_1 \cdot y_2 + z_1 \cdot z_2 \quad (4)$$

Quaternion Hamilton product:

$$\begin{aligned} \mathbf{q}_1 \otimes \mathbf{q}_2 &= (w_1 w_2 + x_1 x_2 + y_1 y_2 + z_1 z_2) \\ &+ (w_1 x_2 + x_1 w_2 + y_1 z_2 + z_1 y_2) i \\ &+ (w_1 y_2 + x_1 z_2 + y_1 w_2 + z_1 x_2) j \\ &+ (w_1 z_2 + x_1 y_2 + y_1 x_2 + z_1 w_2) k \end{aligned} \quad (5)$$

Quaternion conjugates:

The conjugate of  $\mathbf{q}$  is

$$\mathbf{q}^* = w - xi - yj - zk \quad (6)$$

Quaternion inverse:

The inverse of  $\mathbf{q}$  is

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{|\mathbf{q}|} \quad (7)$$

Unit quaternion:

The norm of a quaternion  $\mathbf{q}$  is

$$|\mathbf{q}| = \sqrt{w^2 + x^2 + y^2 + z^2} \quad (8)$$

When  $|\mathbf{q}| = 1$ ,  $\mathbf{q}$  is the unit quaternion and the conjugate of  $\mathbf{q}$  is its inverse.

Rotation of quaternions<sup>[29]</sup>:

In 3D spatial rotation, the rotation axis is assumed to be the unit vector.

$$\mathbf{q} = [\cos\theta, \sin\theta \mathbf{u}] \quad (9)$$

where  $\mathbf{u} = xi + yj + zk$  is the unit quaternion.

The vector  $\mathbf{v}$  (the real part is 0) is rotated by  $2\theta$  around the axis of rotation  $\mathbf{u}$  to obtain  $\mathbf{v}^*$  as

$$\mathbf{v}^* = \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q} - 1 = \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q}^* = \mathbf{v} \odot \mathbf{q} \quad (10)$$

## 2.2 RotatS

When the real part of the quaternion is 0, the quaternion can be expressed as  $\mathbf{v} = xi + yj + zk$ . Such quaternion can be represented as a 3D vector  $\mathbf{v} = (x, y, z)$ .

Similar to the rotation of TeRo<sup>[13]</sup> in 2D space, this paper uses the rotation and scaling of the quaternion with real part 0 in 3D space which are used to complete the TKG completion task. This paper proposes a general model RotatS for TKG completion, which learns the embedding representation of entities, relations, and time in TKG and continuously optimizes the score function, so that correct quaternions receive a higher score and erroneous quaternions have a lower score. In 3D space, the model uses a rotation and scaling based approach where the head entity is rotated and scaled to obtain the tail entity using a combination of time and relations. The general form of the model is as follows.

$$\text{Rot}_{\mathbf{R}, \mathbf{\Gamma}}(\mathbf{H}) = \mathbf{T} \quad (11)$$

where  $\mathbf{H}$  and  $\mathbf{T}$  represent the embedding of head entity and tail entity in 3D space respectively,  $\mathbf{R}$  and  $\mathbf{\Gamma}$  represent the embedding of relation and time in 3D space respectively, and RotS represents the rotation and scaling from head entity  $\mathbf{H}$  to tail entity  $\mathbf{T}$  in 3D space. All embeddings are represented in 3D space, that is

$$\mathbf{H} = (H_x, H_y, H_z) \quad (12)$$

$$\mathbf{T} = (T_x, T_y, T_z) \quad (13)$$

$$\mathbf{R} = (R_x, R_y, R_z) \quad (14)$$

$$\mathbf{\Gamma} = (\Gamma_x, \Gamma_y, \Gamma_z) \quad (15)$$

Through the embedding of entities, relations, and time in 3D space, vector representations of different dimensions can be obtained, thereby obtaining richer semantic information than TeRo. The evolution of head entity to tail entity is performed based on both the representation of relation and time to obtain the semantic association between entity, relation, and time.

It is possible for an event to occur at a timestamp or continuously over a time interval. Rotation axis should be varied to better model different kind of events. Subsections 2.3 and 2.4 present model details for rotating and scaling around the time axis and relational axis, respectively, to address the problem of link prediction for timestamp and time interval events, respectively.

## 2.3 Rotation and scaling around the time axis

The proposed model uses time as the axis of rota-

tion when solving the timestamp problems. The score function of the model is

$$F = \mathbf{H} \odot \mathbf{Q}_T - \mathbf{T} \quad (16)$$

where  $\mathbf{H}$  and  $\mathbf{T}$  are the embedding of the head entity and tail entity in 3D space respectively;  $\odot$  represents the rotation in 3D space. This paper defines  $\mathbf{Q}_T = [\cos \theta_T, \mathbf{u}_T \sin \theta_T]$ , where  $\mathbf{u}_T = \Gamma_x i + \Gamma_y j + \Gamma_z k$  ( $\Gamma_x, \Gamma_y, \Gamma_z$  are the three representations of time in 3D space, and  $\Gamma_x^2 + \Gamma_y^2 + \Gamma_z^2 = 1$ ) is the time axis of the rotation of  $\mathbf{H}$  in 3D space;  $2\theta_T$  is the rotation angle associated with the time embedding.

Because the number of events occurring at a timestamp is limited, this paper uses time as the rotation axis, which can fix the timestamp for inference to narrow the target range of tail entities and thus improve the accuracy of inference. The role of the scoring function is to evolve the head entity  $\mathbf{H}$  to get the tail entity  $\mathbf{T}$  using  $\mathbf{Q}_T$ , which defines a new type of entity relationship and obtains richer semantic association between entities. The goal of this model is that through training, the evolved  $\mathbf{H}$  and  $\mathbf{T}$  of the correct quadruple are getting closer and closer, i. e., the value of  $F$  is getting closer to 0.

However, in the above approach, the effect of relations is not considered. Relations are important and can provide a wealth of information. In static knowledge graphs, the knowledge graph can be complemented by relations alone. Therefore, the paper introduces relation to improve the performance of the model.  $\mathbf{R}^\triangleleft$  is the unitization of relational embedding, which can be represented in 3D space as

$$\begin{aligned} \mathbf{R}^\triangleleft &= (R'_x, R'_y, R'_z) \\ &= (\cos \alpha, \sin \alpha \cos \beta, \sin \alpha \sin \beta) \end{aligned} \quad (17)$$

where  $\alpha, \beta$  are the angles of the relational embedding and are uniformly initialized between  $-\pi$  and  $\pi$ <sup>[29]</sup>. This paper uses  $\circ$  to define Hadmard product between head entities and relations:  $\mathbf{H}^\circ \mathbf{R}^\triangleleft = (H_x, H_y, H_z)^\circ (R'_x, R'_y, R'_z) = (H_x R'_x, H_y R'_y, H_z R'_z)$ . With Hadmard product, this paper incorporates relational information into head entity, thus enriching semantic information. The improved score function obtained in this paper is as follows.

$$F = (\mathbf{H}^\circ \mathbf{R}^\triangleleft) \odot \mathbf{Q}_T - \mathbf{T} \quad (18)$$

Using the above model, the head entity incorporating relational information is rotated in 3D space to obtain tail entity, and the direction of the entity vector changes after the rotation. In order to make the entity vector obtained after evolutions more approximate to the tail entity, this paper performs both the rotation and scaling evolution of the entity, so that the size and direction of the evolved vector are closer to the tail entity

vector, and define  $\mathbf{u}'_r = W_H W_R (\Gamma_x i + \Gamma_y j + \Gamma_z k) = W_H W_R \mathbf{u}_r$ , where  $W_H$  and  $W_R$  are head entity-specific and relation-specific weights respectively. At this time, the value of  $\mathbf{Q}_r$  is as follows.

$$\begin{aligned} \mathbf{Q}'_r &= [\cos \theta_r, \mathbf{u}'_r \sin \theta_r] \\ &= A \times \left[ \frac{\cos \theta_r}{A}, \frac{W_H W_R \sin \theta_r}{A} \mathbf{u}_r \right] \end{aligned} \quad (19)$$

$$A = \sqrt{W_H^2 W_R^2 \sin^2 \theta_r + \cos^2 \theta_r}$$

This paper defines  $\cos \varphi = \frac{\cos \theta_r}{A}$  and  $\mathbf{q} =$

$[\cos \varphi, \mathbf{u}_r \sin \varphi]$ , then if ignoring  $\mathbf{R}^\triangleleft$ , it can obtain:

$$\mathbf{H} \odot \mathbf{Q}'_r = \mathbf{H} \odot (A \times \mathbf{q}) = A^2 \times (\mathbf{q} \otimes \mathbf{H} \otimes \mathbf{q}^*) \quad (20)$$

The multiplication with relation-specificity bias  $W_R$  in Ref. [29] is also used to improve performance. At this point, the score function is

$$F = ((\mathbf{H} \cdot \mathbf{R}^\triangleleft) \odot \mathbf{Q}'_r) \cdot W_R - T \quad (21)$$

Therefore, the rotation axis of the model is  $\mathbf{u}_r$ ; the rotation angle is  $2\varphi$ ; and the scaling factor is  $(W_H^2 W_R^2 \sin^2 \theta_r + \cos^2 \theta_r) W_R$ .

## 2.4 Rotation and scaling around relational axes

In subsection 2.3, this paper discusses the score function when solving the time interval problem. When solving the time interval problem, the relation is used as the rotation axis. The score function is as follows.

$$F = ((\mathbf{H} \cdot \mathbf{R}^\triangleleft) \odot \mathbf{Q}_r) \cdot W_R - T \quad (22)$$

where  $\mathbf{Q}_r = \cos \theta_r + \mathbf{u}_r \sin \theta_r$ ;  $\mathbf{u}_r = W_H W_R (R_x i + R_y j + R_z k) (R_x^2 + R_y^2 + R_z^2 = 1)$ .

An event occurs within a time interval. Here time is a range. It is not reasonable to take a random timestamp from the time interval. Relation is used as the axis of rotation and the head entity incorporating relational information is evolved around the relation axis to obtain the tail entity. The model has the same rotation angle and scaling as the model dealing with timestamp events, and the rotation axis is  $(R_x i, R_y j, R_z k)$ .

## 2.5 Loss function

This paper uses the same loss function as in TeRo<sup>[13]</sup> to optimize the RotatS model, with self-adversarial negative sampling in the loss function:

$$\begin{aligned} Loss &= -\log \sigma(\gamma - y) \\ &\quad - \sum_{i=1}^n p \log \sigma(\hat{y} - \gamma) \end{aligned} \quad (23)$$

where  $\sigma$  is the sigmoid function,  $y$  is the score of the correct quadruple  $(\mathbf{H}, \mathbf{R}, \mathbf{T}, \mathbf{I})$ ,  $\hat{y}$  is the score of the negative sample  $(\hat{\mathbf{H}}, \hat{\mathbf{R}}, \hat{\mathbf{T}}, \hat{\mathbf{I}})$ ,  $\gamma$  is a fixed margin,  $p$  is the ratio of negative training samples to positive training samples. The overall training algorithm

is shown in Algorithm 1.

---

### Algorithm 1 Training of RotatS

---

**Input:** The train set  $TKG = \{\mathbf{H}, \mathbf{R}, \mathbf{T}, \mathbf{I}\}$ ; the number of epoch  $n$ ; the negative sample rate  $p$ .

**Output:** The minimum loss on the train set.

---

```

1:  $s, r, o, t \leftarrow \text{Xavier\_uniform}(), loss_{\min} = 0$  // Initialize
   the embedding vector in 3D space
2: For each  $i \in [1, n]$  do
3:   for  $(s, r, o, t) \in TKG$  do
4:     Constructing negative samples  $D^-$ 
     If  $t$  is time point then calculate score function  $F$ 
5:     by  $((\mathbf{H} \cdot \mathbf{R}^\triangleleft) \odot \mathbf{Q}_r) \cdot W_R - T$ 
6:     end if
     If  $t$  is time period then discretize  $t$  as  $t_2, t_1, \dots$ ,
7:      $t_m$ ; calculate score function  $F$  by  $((\mathbf{H} \cdot \mathbf{R}^\triangleleft) \odot \mathbf{Q}_r) \cdot$ 
      $W_R - T$ 
8:     end if
9:      $Loss = loss_{D^+} + loss_{D^-}$ 
10:    end for
11:     $loss_{\min} = \min(loss_{\min}, loss)$ 
12:    Update parameters  $s, r, o, t$  with Adagrad optimizer
13:  end for
14:  Return  $loss_{\min}$ 

```

---

## 3 Experiments

In experiments, this paper evaluates the performance of the model RotatS on four standard datasets and compared it with some baselines.

### 3.1 Datasets

Frequently used TKGs include ICEWS14, ICEWS05-15, YAGO11k and Wikidata12k. Among them, ICEWS14 and ICEWS05-15 are event-based datasets, both derived from ICEWS<sup>[31]</sup>, corresponding to events in 2014 and events from 2005 to 2015, respectively, both of which record are the timestamps when each event occurs; YAGO11k and Wikidata12k are derived from YAGO<sup>[1]</sup> and Wikidata<sup>[3]</sup>, respectively, and these two datasets record the the time interval in which each event occurs.

### 3.2 Baselines

This paper compares the RotatS model with baselines, including static KGC models and temporal KGC models.

The static KGC models compared in this paper are translation based models and tensor decomposition

based models, including TransE<sup>[5]</sup>, DistMult<sup>[6]</sup>, ComplEX<sup>[8]</sup>, RotatE<sup>[7]</sup> and QuatE<sup>[32]</sup>. These models ignore temporal information when performing KGC.

The temporal KGC models compared in this paper include TTransE<sup>[9]</sup>, TA-TransE<sup>[11]</sup>, TA-DistMult<sup>[11]</sup>, HyTE<sup>[10]</sup>, DE-Simple<sup>[33]</sup>, ATiSE<sup>[12]</sup>, TeRo<sup>[13]</sup>, TempCaps<sup>[14]</sup>, TKGC<sup>[15]</sup> and HTKE<sup>[16]</sup>. These models incorporate temporal information to improve the accuracy of KGC.

### 3.3 Evaluation methodology

This paper evaluates the RotatS model by testing the task on link prediction. The goal of the task is to reason about missing values in  $(*, \mathbf{R}, \mathbf{T}, \mathbf{I})$  and  $(\mathbf{H}, \mathbf{R}, *, \mathbf{I})$ , where  $*$  represents the missing element. For a test quadruple, by substituting either  $H$  or  $T$ , this paper generates candidate quadruplets, calculates scores for all possible quadruplets, and finally ranks these quadruplets in ascending order.

In this experiment, two common evaluation metrics are used; mean reciprocal rank (MRR) and Hits@ $k$ . MRR refers to the average reciprocal rank of all correctly predicted entities. Hits@ $k$  refers to the proportion of correctly predicted entities among the top  $k$  entities.

### 3.4 Experimental setup

This paper evaluates the RotatS model on TITAN Xp, and refers to the experimental setup described in TeRo<sup>[13]</sup>. The model is implemented using Pytorch, and the best model is selected by early stopping on the validation set according to MRR. The experiment chooses embedding size  $k$  as 500, batch size  $b$  as 512, and uses Adagrad as the optimizer. Both real part and imaginary parts of the entity embedding are initialized using uniform initialization.

The time granularity parameters  $u$  and  $thre$  are assigned different values in different datasets. This paper lists the parameters in different datasets<sup>[13]</sup>:  $lr = 0.1$ ,  $gamma = 60$ ,  $thre = 10$  on YAGO11k;  $lr = 0.3$ ,  $gamma = 20$ ,  $thre = 200$  on Wikidata12k;  $lr = 0.1$ ,  $gamma = 110$ ,  $u = 1$  on ICEWS14;  $lr = 0.1$ ,  $gamma = 120$ ,  $u = 2$  on ICEWS05-15.

### 3.5 Complexity

As shown in Table 1, the space complexity and scoring functions between the proposed model and several baselines are compared. In scoring function<sup>[13]</sup>,  $Proj(\cdot)$  represents the temporal projection of the em-

bedding<sup>[11]</sup>; LSTM (long short-term memory) is a neural network model<sup>[10]</sup> that deals with time series problems;  $\rightarrow$  and  $\leftarrow$  represent a time-specific temporal and non-temporal parts of diachronic entity embedding,  $r - 1$  represents the inverse relationship of  $r$ <sup>[11]</sup>;  $D_{KL}(\cdot)$  represents the KL divergence between two Gaussian distributions<sup>[33]</sup>;  $P_{s,t}$ ,  $P_{o,t}$  and  $P_{r,t}$  represent the Gaussian embeddings of  $s$ ,  $r$ , and  $o$  at time  $t$ <sup>[12]</sup>;  $s_t$  and  $o_t$  represent entity embedding at a specific time. In Space Complexity<sup>[13]</sup>,  $m_e$ ,  $m_r$ ,  $m_\tau$  and  $m_{\text{token}}$  represent the number of entities, relations, timestamps, and temporal tokens, respectively;  $d$  is the dimension of the embedding.

Table 1 Comparison of space complexity

| Model       | Scoring function   | Space complexity                        |
|-------------|--|---|
| TransE      | $\ s + r - o\ $  | $O(m_e d + m_r d)$                      |
| TTransE     | $\ s + r + \tau - o\ $   | $O(m_e d + m_r d + m_\tau d)$           |
| HyTE        | $\ Proj(s) + Proj(r) - Proj(o)\ $  | $O(m_e d + m_r d + m_\tau d)$           |
| TA-TransE   | $\ s + LSTM[r, t] - o\ $   | $O(m_e d + m_r d + m_{\text{token}} d)$ |
| TA-DistMult | $s, LSTM[r, t], o$   | $O(m_e d + m_r d + m_{\text{token}} d)$ |
| DE-SimpleE  | $\frac{1}{2}(s \rightarrow, t, o \leftarrow + o \rightarrow, r - 1, s \leftarrow)$ | $O(m_e d + m_r d)$                      |
| ATiSE       | $D_{KL}(P_{s,t} - P_{o,t}, P_{r,t})$   | $O(m_e d + m_r d)$                      |
| TeRo        | $\ s_t + r - \dot{o}_t\ $  | $O(m_e d + m_r d + m_\tau d)$           |
| RotatS      | $RotS_{R,\Gamma}(H) = T$   | $O(m_e d + m_r d + m_\tau d)$           |

## 4 Results

### 4.1 Main results

As shown in Table 2 and Table 3, KGC experiments on the facts at a timestamp are conducted. The prediction accuracy of the performance of RotatS model is listed in table 2 and the benchmark model on the two datasets ICEWS14 and ICEWS05-15 are listed in the table. The results of HTKE, TempCaps, and TKGC are derived from Refs[16], [14] and [15], respectively; the results of other models in the table are derived from TeRo<sup>[13]</sup>. In the experimental results, the RotatS model outperforms all baseline models. It can be seen that the effectiveness of the RotatS model in handling timestamp events. Compared with TeRo, the performance of RotatS model is improves by 2.3% and 4.5% on ICEWS14 and ICEWS05-15, respectively.

Table 2 Performance on the ICEWS14 dataset

| Model           | MRR  | Hit@1 | Hit@3 | Hit@10 |
|-----------------|------|-------|-------|--------|
| TransE          | 28.0 | 9.4   | -     | 63.7   |
| DistMult        | 43.9 | 32.3  | -     | 67.2   |
| ComplEx-N3      | 46.7 | 34.7  | 52.7  | 71.6   |
| RotatE          | 41.8 | 29.1  | 47.8  | 69.0   |
| QuatE           | 47.1 | 35.3  | 53.0  | 71.2   |
| TTransE         | 25.5 | 7.4   | -     | 60.1   |
| HyTE            | 29.7 | 10.8  | 41.6  | 65.5   |
| TA-TransE       | 27.5 | 9.5   | -     | 62.5   |
| TA-DistMult     | 47.7 | 36.3  | -     | 68.6   |
| DE-SimplE       | 52.6 | 41.8  | 59.2  | 72.5   |
| ATiSE           | 55.0 | 43.6  | 62.9  | 75.0   |
| TeRo            | 56.2 | 46.8  | 62.1  | 73.2   |
| TempCaps        | 48.9 | 38.8  | 54.4  | 67.9   |
| TKGC            | 56.1 | 45.8  | 63.1  | 73.8   |
| HTKE            | 56.2 | 47.1  | 61.3  | 72.8   |
| Proposed method | 58.5 | 49.8  | 63.7  | 74.4   |

Table 3 Performance on the ICEWS05-15 dataset

| Model           | MRR  | Hit@1 | Hit@3 | Hit@10 |
|-----------------|------|-------|-------|--------|
| TransE          | 29.4 | 9.0   | -     | 66.3   |
| DistMult        | 45.6 | 33.7  | -     | 69.1   |
| ComplEx-N3      | 48.1 | 36.2  | 53.5  | 72.9   |
| RotatE          | 30.4 | 16.4  | 35.5  | 59.5   |
| QuatE           | 48.2 | 37.0  | 52.9  | 72.7   |
| TTransE         | 27.1 | 8.4   | -     | 61.6   |
| HyTE            | 31.6 | 11.6  | 44.5  | 68.1   |
| TA-TransE       | 29.9 | 9.6   | -     | 66.8   |
| TA-DistMult     | 47.4 | 34.6  | -     | 72.8   |
| DE-SimplE       | 51.3 | 39.2  | 57.8  | 74.8   |
| ATiSE           | 51.9 | 37.8  | 60.6  | 79.4   |
| TeRo            | 58.6 | 46.9  | 66.8  | 79.5   |
| TempCaps        | 52.1 | 42.3  | 57.6  | 70.5   |
| TKGC            | 53.2 | 39.8  | 62.1  | 79.7   |
| HTKE            | 58.1 | 48.7  | 64.5  | 76.3   |
| Proposed method | 63.1 | 52.9  | 70.2  | 80.7   |

As shown in Table 4 and Table 5, KGC experiments on the facts in a certain time interval are conducted, and the results of all models on both the YAGO11k and Wikidata12k datasets are listed in the table. In the experiments, the dataset used for validation has few kinds of relations, which affects the identification of different quadruplets. To increase the variety of relations, this paper extends the relation set to a pair-dual relation set<sup>[13]</sup>. In YAGO11k dataset, tail entity embedding with relation embedding is integrated, the model outperforms

the TeRo model in MRR, Hit@1 and Hit@3; on the Wikidata12k dataset, the performance of RotatS model also improves over the TERO model on Hit@3 and Hit@10.

Table 4 Performance on the YAGO11K dataset

| Model           | MRR  | Hit@1 | Hit@3 | Hit@10 |
|-----------------|------|-------|-------|--------|
| TransE          | 10.0 | 1.5   | 13.8  | 24.4   |
| DistMult        | 15.8 | 10.7  | 16.1  | 26.8   |
| ComplEx-N3      | 16.7 | 10.6  | 15.4  | 28.2   |
| RotatE          | 16.7 | 10.3  | 16.7  | 30.5   |
| QuatE           | 16.4 | 10.7  | 14.8  | 27.0   |
| TTransE         | 10.8 | 2.0   | 15.0  | 25.1   |
| HyTE            | 10.5 | 1.5   | 14.3  | 27.2   |
| TA-TransE       | 12.7 | 2.7   | 16.0  | 32.6   |
| TA-DistMult     | 16.1 | 10.3  | 17.1  | 29.2   |
| ATiSE           | 17.0 | 11.0  | 17.1  | 28.8   |
| TeRo            | 18.7 | 12.1  | 19.7  | 31.9   |
| HTKE            | 25.7 | 17.0  | 25.5  | 40.4   |
| Proposed method | 19.1 | 13.5  | 20.0  | 29.7   |

Table 5 Performance on the Wikidata12k dataset

| Model           | MRR  | Hit@1 | Hit@3 | Hit@10 |
|-----------------|------|-------|-------|--------|
| TransE          | 17.8 | 10.0  | 19.2  | 33.9   |
| DistMult        | 22.2 | 11.9  | 23.8  | 46.0   |
| ComplEx-N3      | 23.3 | 12.3  | 25.3  | 43.6   |
| RotatE          | 22.1 | 11.6  | 23.6  | 46.1   |
| QuatE           | 23.0 | 12.5  | 24.3  | 41.6   |
| TTransE         | 17.2 | 9.6   | 18.4  | 32.9   |
| HyTE            | 18.0 | 9.8   | 19.7  | 33.3   |
| TA-TransE       | 17.8 | 3.0   | 26.7  | 42.9   |
| TA-DistMult     | 21.8 | 12.2  | 23.2  | 44.7   |
| ATiSE           | 28.0 | 17.5  | 31.7  | 48.1   |
| TeRo            | 29.9 | 19.8  | 32.9  | 50.7   |
| Proposed method | 29.6 | 18.7  | 33.2  | 52.7   |

In terms of efficiency, RotatS has the same space complexity as TeRo, TTransE and HyTE. Regarding the consumption of video memory, this paper conducts comparative experiments on the ICEWS14 dataset for several recent models. RotatS with an embedding dimensionality of 500 has a storage size of 1886MiB, whereas TeRo and ATiSE with the same embedding dimensionality have storage sizes of 1026MiB and 1350MiB. Meanwhile, on TITAN Xp device, RotatS

with an embedding dimensionality of 500 is trained on ICEWS14, ICEWS05-15, YAGO11k and Wikidata12k, and each epoch takes 9.1, 49.7 s, 4.2 s and 8.1 s, respectively, however, ATiSE with the same embedding dimensionality takes 7.8 s, 43.0 s, 29.1 s, and 33.7 s on the four datasets, respectively.

#### 4.2 Parameter analysis and model analysis

In this work, the effects of scaling and embedding size on the RotatS model are analyzed.

Fig. 1 shows the link prediction performance on ICEWS14 for different embedding dimensions. As the embedding dimension increases, the prediction performance gradually improves. When the embedding dimension is set to 500, the performance is better and the embedding dimension is not high. A lower embedding dimension can reduce storage space and improve training speed; at the same time, it can also enable the model to be applied on large-scale knowledge graphs.

To analyze the effect of scaling in the RotatS model, this paper compares the link prediction performance of RotatS and RotatS without scaling on ICEWS14.

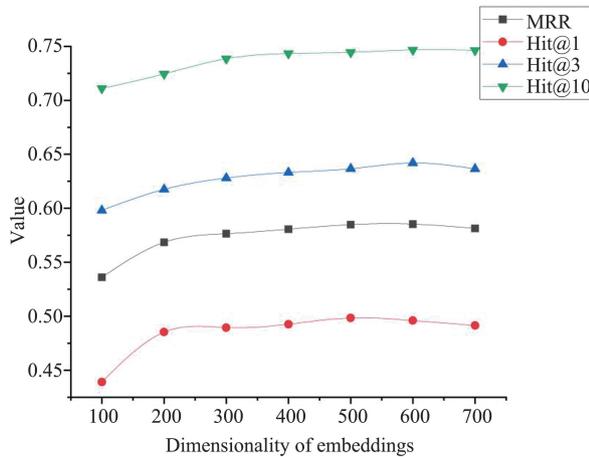


Fig. 1 Effect of dimensionality of embeddings

Fig. 2 shows that scaling can significantly improve the link prediction performance of RotatS. Because scaling can make the size of the head entity vector and the tail entity vector more similar, enhancing the training effect.

#### 4.3 Comparative study

To further test the performance of the proposed TKG completion model in different dimensional spaces, this paper also conducts a series of experiments on rotation and scaling in four dimensional space, and compares this method with the rotation method RotatS in 3D spaces. The model in 4D space is as follows.

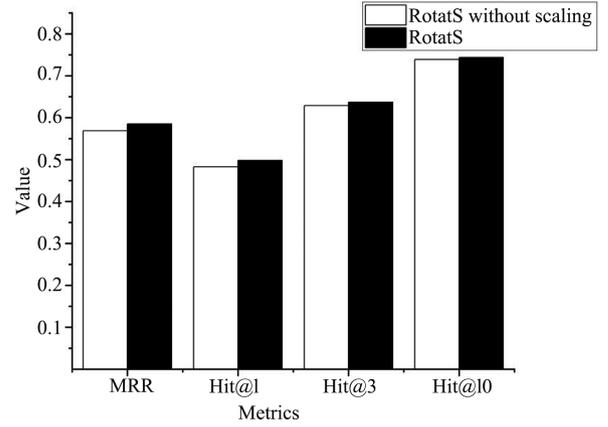


Fig. 2 Effect of scaling

$$(\mathbf{H} \cdot \mathbf{I}^{\triangleleft}) \otimes \mathbf{R}^{\triangleleft} = (\mathbf{T} \cdot \mathbf{I}^{\triangleleft}) \quad (24)$$

$$(\mathbf{H} \cdot \mathbf{I}^{\triangleleft}) \otimes \mathbf{R} = (\mathbf{T} \cdot \mathbf{I}^{\triangleleft}) \quad (25)$$

where  $\mathbf{H}$  and  $\mathbf{T}$  represent the embedding of head entity and tail entity in the 4D space, respectively;  $\mathbf{R}$  represents the embedding of relation in 4D space;  $\mathbf{R}^{\triangleleft}$  and  $\mathbf{I}^{\triangleleft}$  are the unitization of relation embedding and temporal embedding, respectively. The head entity  $\mathbf{H}$  can be represented as  $(H_1, H_2, H_3, H_4)$  in 4D space; the tail entity  $\mathbf{T}$  can be represented as  $(T_1, T_2, T_3, T_4)$  in 4D space; the relation  $\mathbf{R}$  can be represented as  $(R_1, R_2, R_3, R_4)$  in the 4D space; and time  $\mathbf{I}$  can be represented as  $(I_1, I_2, I_3, I_4)$  in 4D space.

The  $\otimes$  denotes the Hamilton product. 4D rotation can be modeled by the Hamilton product<sup>[32]</sup>. When  $\mathbf{R}^{\triangleleft}$  is used, modulus length of  $\mathbf{R}^{\triangleleft}$  is 1, and only rotation is performed in 4D space and no scaling is done; when  $\mathbf{R}$  is used, the modulus length of  $\mathbf{R}$  is not 1, and not only rotation but also scaling is performed in 4D space. In the paper, the time information is incorporated into the head entity and the tail entity by unitizing the Hadamard product between the temporal embedding  $\mathbf{I}$  and the entity embedding. Then the entity quaternions after incorporating the time information are rotated. There are scaled rotations and unscaled rotations. The following are the experimental results of the two methods.

Experiments are performed on two datasets, YAGO11k and ICEWS14, respectively, using the rotation model in 4D space. Under the same experimental conditions as the RotatS model, this paper obtained the results in Table 6. By comparing the experimental results, it can be seen that the rotation model RotatS of quaternion in three-dimensional space performs better in all test cases than that in the 4D space.

## 5 Conclusion

This paper has presented a new TKG completion model named RotatS, which can integrate the semantic

relationship between entities, relations, and time. In the experiments, the rotation and scaling of the head entity to the tail entity is modeled. Compared with previous approaches, the RotatS model has a significant improvement over the baseline model in solving the TKG completion task in sense of prediction accuracy including

MRR, Hit@1, Hit@3 and Hit@10, under similar experimental conditions and space complexity. In the near future, the following research directions will continue to be explored: improving the model by combining different rotation and stretching methods.

Table 6 Rotation in 4D

| Model   | YAGO11K |       |       |        | ICEWS14 |       |       |        |
|---|---------|-------|-------|--------|---------|-------|-------|--------|
|   | MRR     | Hit@1 | Hit@3 | Hit@10 | MRR     | Hit@1 | Hit@3 | Hit@10 |
| $(H \cdot \Gamma^{\triangleleft}) \otimes R^{\triangleleft} = (T \cdot \Gamma^{\triangleleft})$ | 16.9    | 10.9  | 17.2  | 29.4   | 55.3    | 47.2  | 60.1  | 69.8   |
| $(H \cdot \Gamma^{\triangleleft}) \otimes R = (T \cdot \Gamma^{\triangleleft})$                 | 17.2    | 11.1  | 17.8  | 29.3   | 56.2    | 48.7  | 60.2  | 70.2   |

## References

- [1] SUCHANEK F M, KASNECI G, WEIKUM G, et al. Yago: a core of semantic knowledge [ C ] // Proceedings of the 16th International Conference on World Wide Web. Alberta: Association for Computing Machinery, 2007: 697-706.
- [2] BOLLACKER K D, EVANS C, PARITOSH P, et al. Freebase: a collaboratively created graph database for structuring human knowledge [ C ] // Special Interest Group on Management of Data. Vancouver: Association for Computing Machinery, 2008: 1247-1250.
- [3] ERXLEBEN F, MICHAEL G, KRTZSCH M, et al. Introducing wikidata to the linked data web [ C ] // International Semantic Web Conference. Riva del Garda: Springer-Verlag, 2014: 50-65.
- [4] MILLER G A. WordNet: a lexical database for English [ J ]. Communications of the ACM, 1995, 38(11): 39-41.
- [5] BORDES A, USUNIER N, GARCIA-DURAN A, et al. Translating embeddings for modeling multi-relational data [ C ] // Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2013: 2787-2795.
- [6] YANG B, YIH W T, HE X, et al. Embedding entities and relations for learning and inference in knowledge bases [ C ] // International Conference on Learning Representations. Banff: ICLR, 2014: 1-12.
- [7] SUN Z, DENG Z H, NIE J Y, et al. RotatE: knowledge graph embedding by relational rotation in complex space [ C ] // The 7th International Conference on Learning Representations. New Orleans: ICLR, 2019: 1-18.
- [8] TROUILLON T, WELBL J, RIEDEL S, et al. Complex embeddings for simple link prediction [ C ] // Proceedings of the 33rd International Conference on International Conference on Machine Learning. New York: JMLR, 2019: 1-18.
- [9] LEBLAY J, CHEKOL M W. Deriving validity time in knowledge graph [ C ] // Companion of the the Web Conference. Lyon: International World Wide Web Conferences Steering Committee, 2018: 1771-1776.
- [10] DASGUPTA S S, RAY S N, TALUKDAR P. HyTE: hyperplane-based temporally aware knowledge graph embedding [ C ] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: EMNLP, 2018: 2001-2011.
- [11] GARCÍA-DURÁN A, DUMANI S, NIEPERT M. Learning sequence encoders for temporal knowledge graph completion [ C ] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: EMNLP, 2018: 4816-4821.
- [12] XU C, NAYYERI M, ALKHOURY F, et al. Temporal knowledge graph embedding model based on additive time series decomposition [ C ] // International Semantic Web Conference. Paris: ISWC, 2020: 1-8.
- [13] XU C, NAYYERI M, ALKHOURY F, et al. TeRo: a time-aware knowledge graph embedding via temporal rotation [ C ] // Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: COLING, 2020: 1583-1593.
- [14] FU G, ZHAO M, ZHEN H, et al. TempCaps: a capsule network-based embedding model for temporal knowledge graph completion [ C ] // Proceedings of the 6th Workshop on Structured Prediction for NLP. Dublin: SPNLP, 2022: 22-31.
- [15] LINHAI Z, DEYU Z. Temporal knowledge graph completion with approximated gaussian process embedding [ C ] // Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju: COLING, 2022: 12-17.
- [16] PENG H, GANG Z, HONGBO L, et al. Hyperplane-based time-aware knowledge graph embedding for temporal knowledge graph completion [ J ]. Journal of Intelligent and Fuzzy Systems, 2022, 42(6): 5457-5469.
- [17] ZHEN W, ZHANG J, FENG J, et al. Knowledge graph embedding by translating on hyperplanes [ C ] // Proceedings of the 28th AAAI Conference on Artificial Intelligence. Quebec: AAAI Press, 2014: 1112-1119.
- [18] LIN H, YONG L, WANG W, et al. Learning entity and relation embeddings for knowledge resolution [ C ] // Proceedings of the 29th AAAI Conference on Artificial Intelligence. Texas: AAAI Press, 2015: 2181-2187.
- [19] NGUYEN D Q, SIRTS K, QU L, et al. STransE: a novel embedding model of entities and relationships in knowledge bases [ C ] // The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: NAACL, 2016: 460-466.
- [20] JI G, HE S, XU L, et al. Knowledge graph embedding via dynamic mapping matrix [ C ] // Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing. Beijing: ACL-IJCNLP, 2015: 687-696.
- [21] JI G, KANG L, HE S, et al. Knowledge graph completion

- with adaptive sparse transfer matrix [ C ] // Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix; AAAI Press, 2016; 985-991.
- [22] FAN M, ZHOU Q, CHANG E, et al. Transition-based knowledge graph embedding with relational mapping properties [ C ] // Pacific Asia Conference on Language, Information, and Computation. Phuket; PACLIC, 2014; 328-337.
- [23] NICKEL M, TRESP V, KRIEGEL H P. A three-way model for collective learning on multi-relational data [ C ] // International Conference on Artificial Intelligence on Machine Learning. Bellevue; ICML, 2011; 809-816.
- [24] SHI B, WENINGER T. ProjE: embedding projection for knowledge graph completion [ C ] // Proceedings of the 21st AAAI Conference on Artificial Intelligence. San Francisco; AAAI Press, 2017; 1236-1242.
- [25] DETTMERS T, MINERVINI P, STENETORP P, et al. Convolutional 2D knowledge graph embeddings [ C ] // The 32nd AAAI Conference on Artificial Intelligence. San Francisco; AAAI Press, 2017; 1811-1818.
- [26] DAI Q N, TU D N, NGUYEN D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network [ C ] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics; Human Language Technologies. New Orleans; Association of Computational Linguistics, 2018; 327-333.
- [27] XIAOTIAN J, QUAN W, BIN W. Adaptive convolution for multi-relational learning [ C ] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics; Human Language Technologies. Minneapolis; Association of Computational Linguistics, 2019; 978-987.
- [28] GAO H, YANG K, YANG Y, et al. QuatDE: dynamic quaternion embedding for knowledge graph completion [ EB/OL ]. (2021-06-16) [ 2022-11-18 ]. <https://arxiv.org/pdf/2105.09002v2.pdf>.
- [29] GAO C, SUN C, SHAN L, et al. Rotate3D: representing relations as rotations in three-dimensional space for knowledge graph embedding [ C ] // The 29th ACM International Conference on Information and Knowledge Management. New York; Association for Computing Machinery, 2020; 385-394.
- [30] HAMILTON W R. On quaternions, or on a new system of imaginaries in algebra [ J ]. Philosophical Magazine Series, 2009, 31; 511-519.
- [31] LAUTENSCHLAGER J, SHELLMAN S, WARD M. Icwes events and aggregations [ J ]. Harvard Dataverse, 2015, 3.
- [32] ZHANG S, YI T, YAO L, et al. Quaternion knowledge graph embeddings [ C ] // Advances in Neural Information Processing Systems 32; Annual Conference on Neural Information Processing Systems. Vancouver; Association for Computing Machinery, 2019; 2731-2741.
- [33] GOEL R, KAZEMI S M, BRUBAKER M, et al. Diachronic embedding for temporal knowledge graph completion [ C ] // The 34th AAAI Conference on Artificial Intelligence. New York; AAAI Press, 2020; 3988-3995.

**YU Yong**, born in 1995. He is currently pursuing his Ph. D degree in the Institute of Microelectronics of Chinese Academy of Sciences. He received his B. S. and M. S. degrees from Zhengzhou University and University of Chinese Academy of Sciences in 2017 and 2020 respectively. His research interests include knowledge graphs, artificial intelligence, natural language processing.