Data Science and Engineering (2023) 8:135–145
https://doi.org/10.1007/s41019-023-00214-x

RESEARCH PAPER

PosKHG: A Position-Aware Knowledge Hypergraph Model for Link Prediction

Zirui Chen[1] · Xin Wang[1] · Chenxu Wang[1] · Zhao Li[1]

Received: 22 December 2022 / Revised: 11 March 2023 / Accepted: 11 April 2023 / Published online: 5 May 2023

Abstract
Link prediction in knowledge hypergraphs is essential for various knowledge-based applications, including question answering and recommendation systems. However, many current approaches simply extend binary relation methods from knowledge graphs to *n*-ary relations, which does not allow for capturing entity positional and role information in *n*-ary tuples. To address this issue, we introduce PosKHG, a method that considers entities' positions and roles within *n*-ary tuples. PosKHG uses an embedding space with basis vectors to represent entities' positional and role information through a linear combination, which allows for similar representations of entities with related roles and positions. Additionally, PosKHG employs a relation matrix to capture the compatibility of both information with all associated entities and a scoring function to measure the plausibility of tuples made up of entities with specific roles and positions. PosKHG achieves full expressiveness and high prediction efficiency. In experimental results, PosKHG achieved an average improvement of 4.1% on MRR compared to other state-of-the-art knowledge hypergraph embedding methods. Our code is available at https://anonymous.4open.science/r/PosKHG-C5B3/.

Keywords Link prediction · Knowledge hypergraph · Positional embedding · Role

1 Introduction

In *knowledge graphs*, real-world knowledge is represented as triples consisting of a head entity (h), a binary relation (r), and a tail entity (t). Recently, *knowledge hypergraphs* have gained increasing attention as a more general and expressive way of representing knowledge, as they can model nonbinary relations commonly found in real-world scenarios. A significant portion of relations in the Freebase dataset is nonbinary; 61% of relations are nonbinary [1], and more than one-third of entities participate in non-binary relations [27]. Therefore, examining how knowledge hypergraphs can enhance various downstream tasks, such as link prediction and node classification, are significant. However, due to the high cost of storing all correct tuples, knowledge hypergraphs are often incomplete, making it worthwhile to attempt predicting the correctness of hidden tuples based on existing tuples.

N-ary relations, which describe relationships between more than two entities, offer a more nuanced and expressive way to model complex semantics. In Fig. 1, an oval represents a *tuple*, while a circle represents an entity. The entities in the tuple are arranged in a specific order, each occupying a unique *position* and fulfilling a particular *role*. The position represents the order of entities in a tuple. The role is distinct from the entity type and is defined by a particular relation at a specific position, serving as the semantic meaning of the entity in the tuple. For example, in relation to **SportAward**, the roles of entities at positions 1, 2, and 3 are defined as **Award**, **Season**, and **Winner**, respectively. They are represented by the entities Best Scorer, Season 07-08, and LeBron James, indicating that "LeBron James won the best scorer award in the 07-08 season." If LeBron James and Season 07-08 are swapped, the fact represented by the tuple would be inconsistent with reality. This definition was first introduced by Wen et al. [27]. All entities and their respective roles determine the meaning of the tuple. The importance of considering roles and positions in the modeling of knowledge hypergraphs is clear from this example. At the same time, prior work has yet to focus on using position and role information in knowledge hypergraph modeling.

✉ Xin Wang
wangx@tju.edu.cn

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China

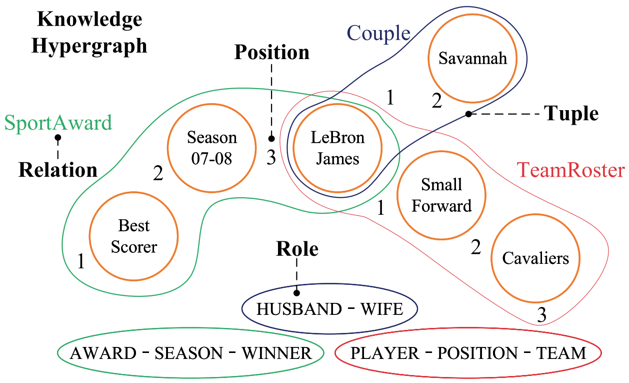


Fig. 1 An example of the knowledge hypergraph for Lebron James is shown. Each relation in the knowledge hypergraph consists of entities with semantic roles assigned to specific positions. These roles may be explicit or implicit, while the positions are always explicit. In the figure, the roles are explicitly indicated

Several previous studies have focused on the link prediction task in knowledge hypergraphs. However, these existing methods [11–30, 1, 19] overlook the importance of positions and roles and continue to utilize binary modeling ideas similar to those used in knowledge graphs. Specifically, these methods embed n-ary relations and entities into a low-dimensional space without considering the specific order of entities and employ these embeddings to assess the plausibility of tuples. For example, m-TransH [27] and RAE [32] both extend the knowledge graph model TransH [25] by projecting entities onto relation-specific hyperplanes for tuple plausibility scoring, but these models have limited expressivity [30, 11], while HypE [1] and GETD [30] extend the knowledge graph model SimplE [11] and TuckER [2], respectively. However, these models entirely ignore position and role semantics [10]. On the other hand, NaLP [9], HINGE [18], and NeuInfer [10] do consider role information. However, they only utilize neural networks to measure tuple plausibility and do not consider the effect of entity position on semantics. To the best of our knowledge, no model currently considers both positional information and role semantics in the context of knowledge hypergraphs.

Therefore, we have identified that, in order to fully and expressively represent knowledge hypergraphs, the following requirements must be met: (1) the complex semantics of tuples in terms of roles and positions should be taken into account during the modeling process, including the semantic relationships among roles, positions, and entity compatibility, and (2) the modeling method should be sufficiently expressive to represent all types of relations. To the best of our knowledge, none of the existing methods fully satisfy these requirements.

In this paper, we focus on each tuple's role and position features in knowledge hypergraphs and propose a Position-award Knowledge HyperGraph embedding model with full expressivity for knowledge hypergraphs. Unlike previous knowledge hypergraph embedding methods, PosKHG introduces a latent space for positions and roles, such that entities with related roles and the same positions have similar representations. Additionally, PosKHG learns a relation matrix for each relation to capture its compatibility with all related entities and devises a scoring function for efficient prediction. The critical innovation of PosKHG is its ability to model knowledge hypergraphs in terms of positions and roles.

The contributions of this paper are as follows:

- PosKHG is a novel knowledge hypergraph embedding model that predicts links in knowledge hypergraphs. It strengthens the emphasis on the roles and positions of entities in n-ary relations. It learns a latent space and a relation matrix of roles to capture semantic relatedness and compatibility.
- We prove that PosKHG is fully expressive for knowledge hypergraphs, which can model all patterns of relations without any limitation.
- Extensive experiments are conducted on six representative datasets, demonstrating that PosKHG achieves state-of-the-art performance on knowledge hypergraph datasets and comparable performance on knowledge graph datasets.

The rest of this paper is organized as follows. In Sect. 3, we introduce the preliminaries. In Sect. 4, we provide a detailed description of PosKHG. Section 5 presents a theoretical analysis of PosKHG's full expressivity and complexity. In Sect. 6, we present the results of our experiments. Finally, we conclude this paper in Sect. 7.

2 Related Work

Our algorithm is conceptually related to previous models in knowledge graphs and recent models in knowledge hypergraphs, which can be classified into two categories.

2.1 Link Prediction in Knowledge Graphs

One popular approach to modeling knowledge graphs is tensor decomposition, exemplified by methods like RESCAL [17] and ComplEx [23]. RESCAL associates knowledge graphs with three-way tensors of head entities, relations, and tail entities and learns entity and relation embeddings to minimize reconstruction error. ComplEx similarly associates each relation with a matrix of head and tail entities, decomposed and learned like RESCAL. However, the main drawback of tensor decomposition methods is that they are limited to modeling a specific arity of relations. At the same time, PosKHG can predict multiple arities of relations simultaneously.

Translation-based methods, such as TransE [4, 24], treat each valid triple as a translation from a head entity to a tail entity through their relation. Subsequently, several improved methods based on TransE have been proposed, including TransH [25], which introduces a relation-specific hyperplane. In TransH, entities are projected onto the relational hyperplane before translation. However, these methods are limited to modeling symmetric relations, while our method, PosKHG, can model any pattern of relations with full expressiveness.

Neural network-based methods model the effectiveness of triples using techniques such as convolutional layers and fully connected layers. For example, ConvKB [16] represents each triple as a three-column matrix fed into these layers to generate an effectiveness score. Nathani [15] proposed a generalized graph attention model as an encoder to capture neighborhood features and applied ConvKB as a decoder. While these methods can be effective, they often come with high time complexity. In contrast, the complexity of our method, PosKHG, is linear in both time and space.

2.2 Link Prediction in Knowledge Hypergraphs

Since binary relations simplify the complexity of real-world facts, some recent studies have tried to represent and predict links in knowledge hypergraphs, primarily through embedding-based methods. These studies represent n-ary facts as tuples with predefined relations and generalize binary relation methods to n-ary cases.

m-TransH [27] and RAE [32] extend the TransH model [25], which is a translation embedding model for binary relations, to the n-ary case. However, these models are not fully expressive and cannot model asymmetric relations. They also do not consider the influence of role and position on tuple semantics. NaLP [9] and HINGE [18] represent n-ary facts as attribute-value pairs and model the associations between these attributes and values. However, these methods assume that the properties of n-ary facts are equally important, which is not always the case in reality. In contrast, our method, PosKHG, uses the tuple form where different entities have different levels of importance, more reflective of real-world scenarios.

RAM [14] and NeuInfer [10] consider incorporating entity role information into embeddings, using tensor decomposition-based and neural network-based methods, respectively, to measure tuple plausibility. However, these models only consider role semantics and do not consider the effect of positional information. To the best of our knowledge, no existing work considers both entity role and positional semantics in knowledge hypergraph modeling. Therefore, we propose PosKHG, which utilizes both role and positional information to improve the performance of knowledge hypergraph link prediction.

3 Preliminaries

This section presents the preliminaries of the knowledge hypergraph and the link prediction task. The notations used throughout the paper are summarized in Table 1.

Definition 1 *(Knowledge Hypergraph)* A knowledge hypergraph is defined as $\mathcal{H} = (\mathcal{E}, \mathcal{R}, \mathcal{T}_O)$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}_O$ is a finite set of entities, relations, and observed tuples, respectively. $t_i = r(\rho_1^r : e_1, \rho_2^r : e_2, ..., \rho_\alpha^r : e_\alpha)$ denotes a tuple where $r \in \mathcal{R}$ is a relation, each $e_i \in \mathcal{E}$ is an entity, i is the position index, each ρ_i^r is the corresponding role of relation r, and α is the non-negative integral arity of the relation r.

After defining the knowledge hypergraph, which follows the definition given by Liu et al. [14], we define the task of link prediction in knowledge hypergraphs.

Definition 2 *(Link Prediction in Knowledge Hypergraphs)* Let $\mathcal{T}$ denote all tuples set, $\mathcal{T}_O \subseteq \mathcal{T}_T \subseteq \mathcal{T}$ indicates the relationship among the set of observed, all ground truth, and all tuples, respectively. The hidden tuples set $\mathcal{T}_H$ is the differences between $\mathcal{T}$ and $\mathcal{T}_O$. Given the observed tuples $\mathcal{T}_O$, the aim of link prediction in knowledge hypergraphs is to predict the labels of the hidden tuples $\mathcal{T}_H$.

Table 1 Notations and explanations

Notation	Explanation
$\mathcal{H}$	Knowledge hypergraph
$e, \mathcal{E}$	Entity and entity set
$r, \mathcal{R}$	Relation and relation set
α	Arity of relation
$t, \mathcal{T}, \mathcal{T}_T, \mathcal{T}_O, \mathcal{T}_H$	Tuple and four kinds of tuple sets
L	Latent space size of role
d	Embedding dimension
ρ^r	Role of relation r
ϕ	Scoring function
σ	Element-wise softmax function
$\mathbf{e}, \mathbf{c}$	Embedding of entity and role
$\mathbf{b}$	Role latent vector
$\mathbf{w}$	Weight vector of role latent vector
$\mathbf{B}$	Basis matrix of relation
$\mathbf{R}$	Relation matrix
$\langle \cdot \rangle$	Multi-linear product

4 The PosKHG Model

As previously mentioned, positions and roles are crucial elements of knowledge hypergraphs, as they help identify entities' semantics and relations. In particular, positions and roles play a crucial role in determining the plausibility of tuples in knowledge hypergraph modeling by considering two key aspects.

- One key aspect of knowledge hypergraph modeling is the semantic relatedness of roles in *n*-ary relational tuples. For example, data from the WikiPeople dataset [9] show that over 80% of the roles found in *n*-ary relations also appear in 2-arity relations. This highlights the importance of considering the shared semantics of different roles in knowledge hypergraph modeling.
- Another factor contributing to the plausibility of tuples in knowledge hypergraph modeling is the compatibility between positions, roles, and entities, e.g., in Fig. 1, the role of "Player" and the corresponding entity of "LeBron James" interact with the entities of "Small Forward" and "Cavaliers" to affect the overall plausibility of the tuple. In other words, the compatibility between different elements within a tuple is crucial for accurately modeling knowledge using hypergraphs.

To address these issues, we propose a method called PosKHG that models knowledge hypergraphs at the role and position level, allowing for the capture of the semantic relatedness between roles and positions through a latent space. A relation matrix is also used to represent the compatibility between roles, positions, and all relevant entities. To measure the plausibility of tuples, we use a multi-linear product, which allows for full expressiveness. The overall structure of PosKHG is depicted in Fig. 2.

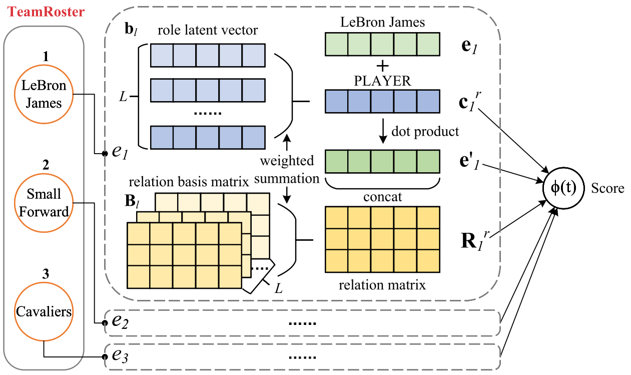


Fig. 2 Overview of PosKHG. Each entity generates an entity embedding, combining it with the role embedding to integrate the role semantics. Then, implements the incorporation of the positional semantics through concatenation operation. The relation matrix is produced by combining the basis matrices of a relation. Finally, entity embedding, role embedding, and relation matrix of each entity are fed into the scoring function to calculate the confidence score of the tuple

4.1 Latent Space for Roles and Positions

Since one entity may correspond to multiple positions and roles in a knowledge hypergraph dataset, such as the entity LeBron James in Fig. 1, the multi-embedding mechanism [22] is designed for entities and maps each entity $e_i \in \mathcal{E}$ to multiple embeddings. Let $\mathbf{e}_i \in \mathbb{R}^{m \times d}$ denote entity embedding, m be the layers of multi-embedding, d be the embedding dimension.

Inspired by sharing feature information of training examples in machine learning [26, 35, 28], in order to utilize semantic information about the positions and roles of the entities, a latent space is built for roles with L role latent vectors $\mathbf{b}_l \in \mathbb{R}^d, l = 1, 2, ..., L$. First of all, the role embedding $\mathbf{c}_i^r$ is computed by a combination of role latent vectors:

$$\mathbf{c}_i^r = \sum_{l=1}^{L} \mathbf{b}_l \cdot \sigma(\mathbf{w}_i^r)[l] \tag{1}$$

where $\mathbf{w}_i^r \in \mathbb{R}^L$ is the weight vector of the role latent vector, known as the role weights. As a result, semantic relatedness is parameterized implicitly by role weights, while the weight vector needs to be normalized by the element-wise Softmax function σ for all l and $l' \leq L$:

$$\sigma(\mathbf{w}_i^r)[l] = \exp(\mathbf{w}_i^r[l]) / \sum_{l'=1}^{L} \exp(\mathbf{w}_i^r[l']) \tag{2}$$

After obtaining the role embedding, it further assigning various semantics to entities that at different positions. To be specific, the entity embedding and the role embedding are multiplied to obtain an embedding $\mathbf{e}'_i = \mathbf{e}_i \cdot \mathbf{c}_i^r$ that incorporates the role semantics. Moreover, the i-th positional semantics is combined by the concatenation function cat:

$$\mathbf{e}'_i = (\mathbf{e}_i^1, cat(\mathbf{e}_i^2, m \cdot d/\alpha), ..., cat(\mathbf{e}_i^\alpha, m \cdot d \cdot (\alpha - 1)/\alpha)) \tag{3}$$

where $cat(\mathbf{v}, x)$ shifts vector $\mathbf{v}$ to the left by x steps.

4.2 Relation Matrix

The relations in knowledge hypergraphs consist of entities at different positions and corresponding roles. Figure 3 illustrates the latent space for roles. To measure the degree of compatibility among the positions, roles, and all participated entities, the roles at each position in the relation are learned with a relation matrix. For a relation $r \in \mathcal{R}$, the relation matrix for the role at the i-th position is represented by $\mathbf{R}_i^r \in \mathbb{R}^{\alpha \times m}$, where the j-th row $\mathbf{R}_i^r[j, :]$

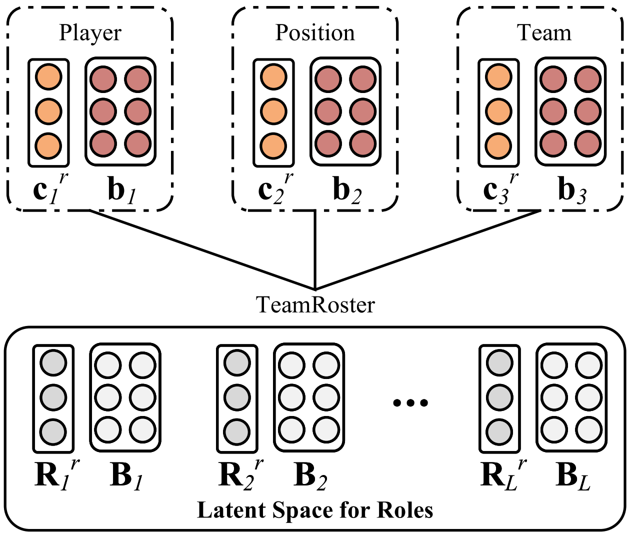


Fig. 3 Latent space for roles and relation matrices in PosKHG, with the multiplicity of entity embeddings $m = 2$ (in example of relation TeamRoster with roles (player, position, and team))

denotes the compatibility with multi-embedding of the j-th position entity. With a designed latent space of positions and roles, the relation matrix can be learned as follows for all $i \leq \alpha$:

$$\mathbf{R}_i^r = \sum_{l=1}^{L} \sigma(\mathbf{w}_i^r)[l] \cdot \sigma(\mathbf{B}_l) \tag{4}$$

where $\mathbf{B}_l \in \mathbb{R}^{\alpha \times m}$ is the basis matrix of relation linked with latent vector of role $\mathbf{b}_l$ in the latent space. The entire basis matrix is also normalized by σ. The basis relation matrix $\mathbf{B}_l$ is aligned with the latent role vector $\mathbf{b}_l$, which is used to compute for role embeddings and relation matrices.

4.3 Scoring Function

The scoring function employs a multi-linear product approach to calculate the confidence of the knowledge hypergraph tuple, which can effectively improve the performance and introduce fewer parameters, making the training more efficient. For each tuple $t_i = r(\rho_1^r : e_1, \rho_2^r : e_2, ..., \rho_\alpha^r : e_\alpha)$, the score of tuples is calculated by the following equation:

$$\phi(t) = \sum_{i=1}^{\alpha} \langle \mathbf{c}_i^r, \mathbf{R}_i^r[1,:]\mathbf{e}_1, ..., \mathbf{R}_i^r[\alpha,:]\mathbf{e}_\alpha \rangle \tag{5}$$

where $\mathbf{R}_i^r[1,:]\mathbf{e}_1$ captures the compatibility between the role ρ_i^r and i-th entity e_i, i.e., the multi-embedding of e_i is weighted by the elements of $\mathbf{R}_i^r[1,:]$. Each summation term of the multi-linear product is the compatibility of the entity with the corresponding role at a different position.

4.4 Model Training

Generally, the knowledge hypergraph only provides positive examples, while negative examples need to be sampled by some way. Based on the scoring function designed above, the training loss and the learning target of the model are designed in the following way. For each positive tuple $t \in \mathcal{T}_O$, the negative samples are obtained by replacing the entity linked with ρ_i^r. The strategy generalizes from the ones in the binary case:

$$\mathcal{S}_t^{(i)} = \bigcup_{i=1}^{\alpha} \{ r(e_1, ..., \hat{e}_i, ..., e_\alpha) \notin \mathcal{T}_O \mid \hat{e}_i \in \mathcal{E}, \hat{e}_i \neq e_i \} \tag{6}$$

Furthermore, an instantaneous multi-class log-loss is adopted and a optimizer of an empirical risk is formulated as follows:

$$\min_{\mathbf{e}_i \in E, \mathbf{b}_i \in B, \mathbf{w}_i^r \in W, \mathbf{B}_i \in \mathcal{B}} \sum_{t \in \mathcal{T}_O} \sum_{i=1}^{\alpha} -\log \Big[\exp(\phi(t)) / (\exp(\phi(t)) + \sum_{t' \in \mathcal{S}_t^{(i)}} \exp(\phi(t'))) \Big] \tag{7}$$

where the set E, B, W, and $\mathcal{B}$ contains all elements of $\mathbf{e}_i$, $\mathbf{b}_i$, $\mathbf{w}_i^r$, and $\mathbf{B}_i$, respectively, the Softmax loss guarantees that exactly one correct sample is learned among the candidates.

Algorithm 1 is the training process of PosKHG. For each tuple sampled from a knowledge hypergraph, its negative sample is obtained at first. Next, the embeddings and the relation matrix are computed. Then, the confidence score of this sampled tuple is calculated. Finally, PosKHG is trained in mini-batch to minimize the above empirical risk formulation.

Algorithm 1 Training procedure for PosKHG

```
Require: Observed tuples $\mathcal{T}_O$, iteration count $n_{\text{iter}}$, mini-batch size $m_b$, latent
    space size $L$
Ensure: Role embedding, entity embedding, relation matrices
for $t = 1, ..., n_{\text{iter}}$ do
    Sample a mini-batch $\mathcal{T}_{\text{batch}} \subseteq \mathcal{T}_O$ of size $m_b$
    for each tuple $t \in \mathcal{T}_{\text{batch}}$ do
        Construct negative samples for tuple $t$
        $\mathbf{c}_i^r \leftarrow$ compute role embeddings using (1)
        $\mathbf{e}_i' \leftarrow$ compute entity embeddings using (3)
        $\mathbf{R}_i^r \leftarrow$ compute relation matrices using (4)
        Update learnable parameters w.r.t. gradients based on the whole
    objective in (7)   =0
```

5 Theoretical Analysis

Table 2 summarizes the role-aware and position-aware properties, expressiveness, and the time and space complexity of existing n-ary relational approaches.

The PosKHG model is fully expressive, indicating that the model can correctly learn any valid n-ary relation in the knowledge hypergraph without being restricted to a specific pattern of relations. Given any ground truth tuples in the knowledge hypergraph, at least one embedding assignment of the model can correctly separate valid tuples from invalid ones. Furthermore, the PosKHG model can achieve a linear time and space complexity. Its embedding dimension constraint is presented in Theorem 1, and the complexity analysis is conducted.

Theorem 1 *For any ground truth over entities $\mathcal{E}$ and relations $\mathcal{R}$ of the knowledge hypergraph containing $\eta \geq 1$ ground truth tuples, there exists a PosKHG model with the embedding dimension $d = \eta$, the multiplicity of entity embedding $m = \max_{r \in \mathcal{R}} \alpha$, and the latent space size $L = \eta$, which accurately represents the ground truth tuple.*

Proof Let $\mathcal{T}_T$ be the set of all ground truth tuples in the knowledge hypergraph with $\rho = |\mathcal{T}_T|$. Then, the statement of Theorem 1 is equivalent to assigning parameters entity embeddings E, role basis vectors B, role weights W, and relation basis matrices $\mathcal{B}$ to PosKHG. Under the conditions of embedding dimension $d = \eta$, multiplicity of entity embeddings $m = \max_{r \in \mathcal{R}} \alpha$, and latent space size $L = \eta$, the scoring function can be expressed as follows:

$$\phi(t) = \begin{cases} > 0, & \text{if } t \in \mathcal{F}^{\text{all}} \\ = 0, & \text{if } t \notin \mathcal{F}^{\text{all}} \end{cases}, \text{ for } t{:} = \{\rho_1^r{:}e_1, ..., \rho_\alpha^r{:}e_\alpha\} \quad (8)$$

when $\eta \geq 1$, for each entity $e \in \mathcal{E}$ with multiple embeddings $\mathbf{e} \in \mathrm{R}^{m \times d}$, $\mathbf{e}[i, j]$ is set to 1 if the entity e involves with the i-th role of the j-th tuple in $\mathcal{T}_T$, and to 0 otherwise. As for the latent space, an identity matrix $\mathbf{I}_L$ is concatenated by the role latent vectors $[\mathbf{b}_1, ..., \mathbf{b}_l]$. The form of the relation basis matrix is $\mathbf{B}_i = [\mathbf{I}_\alpha, 0] \in \{0, 1\}^{\alpha \times m}$. Since an identity matrix

Table 2 A comparison of knowledge hypergraph models. The role-aware property considers the semantic relatedness between different roles and the compatibility between roles and entities. The position-aware property involves considering the semantic information between entities and the compatibility between relations and entities. N/A terms are not definite in literature. Additionally, m_e and m_r represent the number of entities and relations in the model. In contrast, m_a is the maximum arity of the knowledge hypergraph, and d is the embedding dimensionality

Model	Role-aware	Position-aware	Expressive	$\mathcal{O}_{\text{time}}$	$\mathcal{O}_{\text{space}}$
m-TransH	×	×	×	$\mathcal{O}(d)$	$\mathcal{O}(m_e d + 2m_r d_\alpha)$
RAE	×	×	×	$\mathcal{O}(d^2)$	$\mathcal{O}(m_e d + 2m_r d_\alpha)$
NaLP	×	×	N/A	$\mathcal{O}(d^2)$	$\mathcal{O}(m_e d + m_r m_a d_\alpha)$
HINGE	×	×	N/A	$\mathcal{O}(d^2)$	$\mathcal{O}(m_e d + m_r m_a d_\alpha)$
NeuInfer	×	×	N/A	$\mathcal{O}(d^2)$	$\mathcal{O}(m_e d + m_r m_a d_\alpha)$
HypE	×	✓	✓	$\mathcal{O}(d)$	$\mathcal{O}(m_e d + m_r d_\alpha)$
RAM	✓	×	✓	$\mathcal{O}(d)$	$\mathcal{O}(m_e d + Lm_r d_\alpha)$
PosKHG	✓	✓	✓	$\mathcal{O}(d)$	$\mathcal{O}(m_e d + Lm_r m_\alpha)$

is a group of latent vectors for R^{η}, the role weights $\{\mathbf{W}_i^r\}$ can be assigned to satisfy that $\mathbf{c}_i^r[j] = 1$ if the relation r involves with the j-th tuple in $\mathcal{T}_T$, and $\mathbf{c}_i^r = 0$ otherwise. Then, the confidence score of the j-th ground truth tuple can be calculated by PosKHG through the following equation:

$$\phi(t) = \sum_{i=1}^{\alpha} \langle \mathbf{c}_i^r, \mathbf{R}_i^r[1, :]\mathbf{e}_1, ..., \mathbf{R}_i^r[\alpha, :]\mathbf{e}_\alpha \rangle \tag{9}$$

and each summation term is equal to 1, the score for t is $\phi(t) = \alpha > 0$

As for $\phi(t) = 0$, assume there exists a false tuple $t \notin \mathcal{T}_T$, $\phi(t) > 0$. Based on this assumption, there is at least one position j to ensure that $\mathbf{c}_i^r[j] = 1$ and the j-th elements of $\mathbf{R}_i^r[1, :]\mathbf{e}_1, ..., \mathbf{R}_i^r[\alpha, :]\mathbf{e}_\alpha$ are all equal to 1. However, this can only happen when entities $e_1, ..., e_\alpha$ and relation r appear in the j-th tuple of $\mathcal{T}_T$ simultaneously, and then, $t \in \mathcal{T}_T$ which contradicts the initial assumption. So that when $t \notin \mathcal{T}_T$, $\phi(t) = 0$. □

For the time complexity, since our scoring function uses a multi-linear product, the linear time complexity is $\mathcal{O}(d)$. For space complexity, since the arity of relations in the knowledge hypergraph is rarely higher than 6 (as shown in Table 3), the assignment of parameter m will not exceed 3. If let m_α be the maximum arity of relation in the knowledge hypergraph, m_e be the number of entities, m_r be the number of relations, the parameters spent on the role latent vector, the basis matrix of relation, and the role weight vector are at most $\mathcal{O}(m_e d + L m_r m_\alpha + Ld + Lmm_\alpha) = \mathcal{O}(m_e d + L m_r m_\alpha)$. Thus, the PosKHG model remains linear in both time and space.

6 Experiments

The performance of PosKHG was evaluated on two types of benchmarks. Section 6.1 describes the experimental setups, including the datasets and baselines used. In Sect. 6.2, we present the results of experiments designed to predict hidden tuples or hidden triples.

6.1 Experiment Settings

6.1.1 Datasets

The experiments on link prediction were conducted on six datasets. The knowledge hypergraph dataset JF17K was proposed by Wen et al. [27], while FB-AUTO was proposed by Fatemi et al. [1]. As no validation set was proposed for JF17K, we randomly selected 20% of the train set as validation. Four standard knowledge graph benchmarks, i.e., WN18, FB15k, WN18RR, and FB15k-237 were used for link prediction in knowledge graphs. The detailed statistics of the datasets are summarized in Table 3.

6.1.2 Baselines

For link prediction in knowledge hypergraphs, we compare PosKHG with the state-of-the-art approaches, including RAE [32], NaLP [9], HINGE [18], NeuInfer [10], HypE [1], and RAM [14]. In addition, GETD [30] can only model single-arity knowledge hypergraphs and therefore is not included in the comparison. As for link prediction in knowledge graphs, we compared PosKHG with several baselines, including TransE [3], DistMult [24], ComplEx [23], SimplE [11], RotatE [21], TuckER [2], HAKE [31], DualE [5], AutoSF [34], and ComplExRP [6].

6.1.3 Evaluation Metrics

Two evaluation metrics were employed to compare the performance of different link prediction methods: mean reciprocal rank (MRR) and Hit@K, where H@K is in %, and all results in Sect. 6.2 are rounded. Two metrics above are measured by ranking a test tuple t within a set of replaced tuples. For each tuple in the test set and each position i in

Table 3 Dataset statistics

Dataset	# entities	# relations	# train	# valid	# test	# arity = 2	# arity = 3	# arity = 4	# arity = 5	# arity = 6
FB15k	14, 951	1345	483, 142	50, 000	59, 071	592, 213	0	0	0	0
WN18	40, 943	18	141, 442	5000	5000	151, 422	0	0	0	0
FB15k-237	14, 541	237	272, 115	17, 535	20, 466	310, 116	0	0	0	0
WN18RR	40, 493	11	86, 835	3034	3134	93, 003	0	0	0	0
JF17K	29, 177	327	77, 733	–	24, 915	56, 332	34, 550	9509	2230	37
FB- AUTO	3388	8	6778	2255	2180	3786	0	215	7212	0

Table 4 Results of Link Prediction on Knowledge Hypergraph Datasets

Model	JF17K				FB-AUTO			
	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
RAE	0.396	0.312	0.433	0.561	0.703	0.614	0.764	0.854
NaLP	0.310	0.239	0.334	0.450	0.672	0.611	0.712	0.774
HINGE	0.473	0.397	0.490	0.618	0.678	0.765	0.706	0.765
NeuInfer	0.451	0.373	0.484	0.604	0.737	0.700	0.755	0.805
HypE	0.507	0.421	0.550	0.669	0.804	0.774	0.824	0.856
RAM	0.539	0.463	0.573	0.690	0.830	0.803	0.851	0.876
PosKHG⊛ (Ours)	0.541	0.462	0.575	0.695	0.843	0.810	0.857	0.885
PosKHG⋇ (Ours)	0.521	0.442	0.551	0.664	0.823	0.794	0.840	0.862
PosKHG (Ours)	**0.545**	**0.469**	**0.582**	**0.706**	**0.856**	**0.821**	**0.876**	**0.895**

Bolded and underlined are the highest and second highest performance results under the current evaluation metric, respectively

the tuple, $|\mathcal{E}| - 1$ replaced tuples are generated by replacing the entity e_i with each entity in $\mathcal{E} \backslash \{e_i\}$.

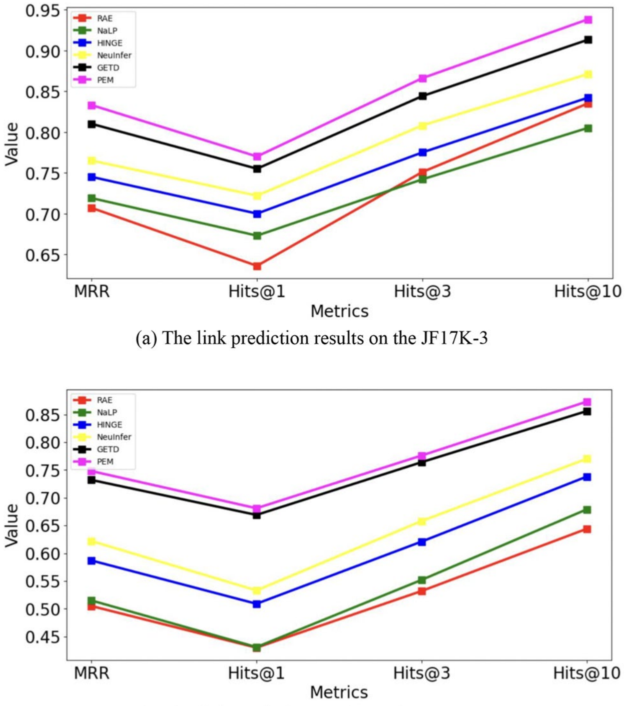
6.2 Results

6.2.1 Link Prediction in Knowledge Hypergraphs

Table 4 shows that PosKHG improves the MRR on the FB-AUTO dataset by at most 2.6%. When modeling knowledge hypergraphs, RAE, NaLP, and HINGE ignore positional and role semantics. RAE is based on generalizing the TransH translation model for knowledge hypergraphs. However, it is not fully expressive and can only model symmetric relations, leading to lower prediction performance than state-of-the-art models. NaLP and HINGE split tuples into primary tuple and auxiliary key-value pair attributes, which ignore the semantic information of positions and roles, resulting in lost information and lower prediction performance.

HypE is a generalization of the SimplE model, which variates the position differences but not the role. The experimental results of RAM (only considering roles) outperform that of HypE, which further illustrates the importance of role semantics in link prediction. While NeuInfer and RAM model the knowledge hypergraph using neural networks and tensor decomposition-based methods, respectively, which consider the difference in role semantics but not the positional information. The lack of utilizing positional information causes the worse experimental results of PosKHG, which fully justifies the significance of positional information for knowledge hypergraph modeling.

We conducted an ablation study that separately excluded position and role information and labeled the results as PosKHG⊛ and PosKHG⋇, respectively, to demonstrate the importance of position and role information in link prediction tasks. As shown in Table 4, the experimental results for PosKHG⊛ are still better than those for RAM, which only considers role semantics, and the results for PosKHG⋇ are still better than those for HypE, which only considers position semantics. However, the performance of both variants is inferior to that of PosKHG. This supports our conclusion that both position and role information are essential for knowledge hypergraph modeling.



(a) The link prediction results on the JF17K-3

(b) The link prediction results on the JF17K-4

Fig. 4 Link Prediction in Knowledge Hypergraph with Fixed Arity on JF17K Dataset

6.2.2 Link Prediction with Different Arities

As shown in Fig. 4, to compare PosKHG with other models, we conducted experiments on knowledge hypergraphs with fixed arity using the JF17K-3 and JF17K-4 datasets, subsets of JF17K containing tuples with specific arities of relations.

Among the translational models, we only include RAE, an improved version of m-TransH. As baselines, we use the tensor decomposition models GTED and the neural network models NaLP, HINGE, and NeuInfer.

The performance on knowledge hypergraph data with fixed arity is shown in Table 5. The tensor decomposition model generally outperforms other models. This is likely due to their intense expressiveness. Our proposed PosKHG consistently achieves state-of-the-art performance on all benchmark datasets due to its consideration of both position and role in its design.

Table 5 presents the results of directly predicting tuples of different arities after training on the entire knowledge hypergraph dataset. The RAM model considers role semantics but not positions, while GETD, an extension of the TuckER model, only considers positions but not roles.

PosKHG performs the best on all arities of relations in the FB-AUTO dataset, and on the JF17K dataset, it performs the best on high-arity relations. This is likely due to low-arity data noise during the training process for high-arity predictions. PosKHG improves by an average of 4.1% on all arities compared to RAM, demonstrating the importance

Table 5 Results of Link Prediction on different arities of knowledge hypergraph datasets

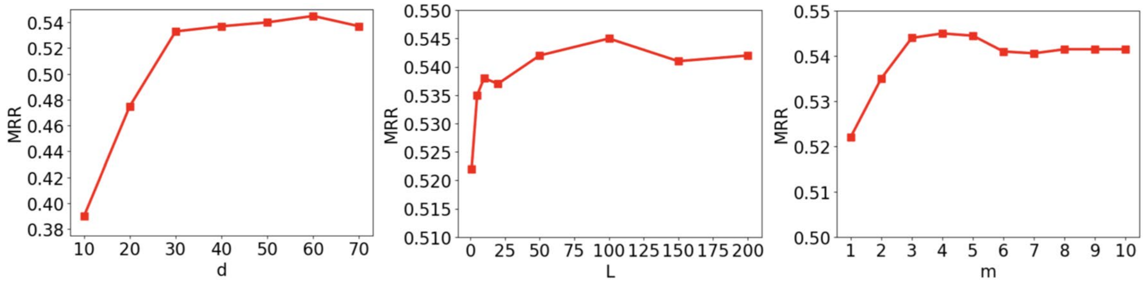
Model	JF17K					FB-AUTO		
	2	3	4	5	6	2	4	5
GETD	**0.339**	**0.583**	**0.751**	0.746	0.350	0.524	0.237	0.786
RAM	0.337	0.578	0.736	0.805	0.697	0.557	0.456	0.904
PosKHG (Ours)	0.334	0.577	0.739	**0.813**	**0.708**	**0.572**	**0.477**	**0.912**

Bolded values indicate the highest performance results under the current evaluation metric

Table 6 Results of link prediction on knowledge graph datasets

Model	WN18			FB15k			WN18RR			FB15k-237		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
TransE	0.495	0.113	0.943	0.463	0.297	0.749	0.226	–	0.501	0.294	–	0.46
DistMult	0.822	0.728	0.936	0.654	0.546	0.824	0.430	0.390	0.490	0.241	0.155	0.419
ComplEx	0.941	0.939	0.947	0.727	0.660	0.838	0.440	0.410	0.510	0.247	0.158	0.428
SimplE	0.942	0.939	0.947	0.727	0.838	0.660	–	–	–	–	–	–
RotatE	0.949	0.944	0.959	0.797	0.746	0.884	0.476	0.428 – –	0.571	0.338	0.241	0.533
TuckER	**0.953**	**0.949**	0.958	0.795	0.741	0.892	–	–	–	–	–	–
HAKE	–	–	–	–	–	-	**0.497**	**0.452**	**0.582**	0.346	0.250	0.542
DualE	0.951	0.945	**0.961**	0.790	0.734	0.881	–	–	–	–	–	–
AutoSF	0.952	0.947	**0.961**	**0.853**	**0.821**	**0.910**	0.490	0.451	0.567	0.360	0.267	0.552
ComplExRP	–	–	–	–	–	–	0.488	0.443	0.578	**0.388**	**0.298**	**0.568**
PosKHG (Ours)	0.943	0.940	0.949	0.801	0.751	0.877	0.496	0.449	0.577	0.349	0.248	0.544

Bolded and underlined are the highest and second highest performance results under the current evaluation metric, respectively



(a) MRR v.s. ***d***

(b) MRR v.s. ***L***

(c) MRR v.s. ***m***

Fig. 5 Effects of embedding dimensionality *d*, latent space size *L*, and multiplicity of entity embeddings *m* on the testing MRR on JF17K dataset

of considering both positional and role information for link prediction tasks in knowledge hypergraphs.

6.2.3 Link Prediction in Knowledge Graphs

PosKHG achieves the second-best results on MRR and Hit@1 metrics and performs comparably on FB15k. These results demonstrate that while PosKHG may not achieve the optimal performance on binary datasets, it can achieve performance similar to binary relation models (Table 6). This shows that its design, which considers both position and (implicit) role, equally applies to knowledge graphs.

6.2.4 Sensitivity of Hyperparameters

Figure 5 further investigates the influence of critical hyperparameters on the JF17K dataset, including the embedding dimensionality (d), the latent space size (L), and the multiplicity of entity embeddings (m). Since the full expressiveness of PosKHG guarantees its learning capacity, it consistently performs well when d is greater than 30 (Fig. 5a). As for Fig. 5b, a small latent space size ($L \geq 5$) is sufficient for robust performance, which confirms the effectiveness of role semantic relatedness in knowledge hypergraphs. Figure 5(c) shows a peak point with the multiplicity of entity embeddings providing the best coverage for entity semantics when $m = 3$ or 4. Larger values of m lead to overfitting and intractable learning. Existing bilinear models also choose small values of m, such as $m = 3$ in DistMult and SimplE. Therefore, an appropriate value for m is 3 to 5, while the optimal values for the embedding dimensionality and latent space size depend on the scale of the dataset.

7 Conclusion

In this paper, we propose a link prediction model PosKHG for knowledge hypergraphs, which learns the embedding representation from both role and position levels. Leveraging the latent space for entity semantic relatedness of role and position and relation matrix for entity compatibility achieves precise accuracy for link prediction, full expressiveness, and more generalized modeling of knowledge hypergraphs. The experimental results on both knowledge hypergraph datasets and four knowledge graph datasets demonstrate the superiority and robustness of PosKHG.

Acknowledgements This work is supported by the National Key R &D Program of China (2020AAA0108504) and National Natural Science Foundation of China (61972275).

Author Contributions All the authors listed in this manuscript have made substantial contributions to this article and signed appropriately.

Funding This study was funded by the National Key R &D Program of China (2020AAA0108504) and National Natural Science Foundation of China (61972275).

Availability of data and materials Available.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate All authors agreed to participate.

References

1. Bahare F, Perouz T, David V, Poole D (2020) Knowledge hypergraphs, prediction beyond binary relations In: IJCAI
2. Balazevic I, Allen C, Hospedales T (2019). TuckER: tensor factorization for knowledge graph completion. In: EMNLP, pp 5188–5197
3. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: NeurIPS
4. Bordes A, Usunier N, GarciaDuran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multirelational data. In: Proceedings of the 26th international conference on neural information processing systems, pp 2787–2795
5. Cao Z, Xu Q, Yang Z, Cao X, Huang Q (2021) Dual quaternion knowledge graph embeddings. In: Proceedings of the AAAI conference on artificial intelligence 35(8):6894–6902
6. Chen Y, Minervini P, Riedel S, Stenetorp P (2021) Relation prediction as an auxiliary training objective for improving multi-relational graph representations. arXiv preprint arXiv:2110.02834
7. Ding B, Wang Q, Wang B, Guo L (2018) Improving knowledge graph embedding using simple constraints. In: Proceedings of the 56th annual meeting of the association for computational linguistics, pp 110–121
8. Ebisu T, Ichise R (2018) TorusE: knowledge graph embedding on a lie group. In: Proceedings of the 32nd AAAI conference on artificial intelligence, pp 1819-1826
9. Guan S, Jin X, Wang Y, Cheng X (2019) Link prediction on n-ary relational data. In: Proceedings of the 2019 world wide web conference, pp 583–593
10. Guan S, Jin X, Guo J, Wang Y, Cheng X (2020) NeuInfer: knowledge inference on n-ary facts. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 6141–6151

11. Kazemi SM, Poole D (2018) SimplE embedding for link prediction in knowledge graphs. In: NeurIPS
12. Lacroix T, Usunier N, Obozinski G (2018) Canonical tensor decomposition for knowledge base completion. In: ICML
13. Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015b) Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the 29th AAAI conference on artificial intelligence, pp 2181–2187
14. Liu Y, Yao Q, Li Y (2021) Roleaware modeling for n-ary relational knowledge bases. arXiv preprint arXiv:2104.09780
15. Nathani D, Chauhan J, Sharma C, Kaul M (2019) Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 4710–4723, Florence, Italy
16. Nguyen DQ, Nguyen TD, Nguyen DQ, Phung D (2018) A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of the 16th annual conference of the north American chapter of the association for computational linguistics: human language technologies, pp 327–333
17. Nickel M, Tresp V, Kriegel H-P (2011) A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th international conference on machine learning, pp 809–816
18. Paolo R, Dingqi Y, Philippe C (2020) Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In: Proceedings of the web conference 2020, pp 1885–1896
19. Peng Y, Choi B, Xu J (2021) Graph learning for combinatorial optimization: a survey of state-of-the-art. Data Sci. Eng. 6(2):119–141
20. Rossi A, Barbosa D, Firmani D, Matinata A, Merialdo P (2021) Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans. Knowl. Discov. Data 15(2):1-49
21. Sun Z, Deng Z-H, Nie J-Y, Tang J (2019) RotatE: knowledge graph embedding by relational rotation in complex space. In: ICLR, Article 14.
22. Tran HN, Takasu A (2019) Analyzing knowledge graph embedding methods from a multi-embedding interaction perspective. arXiv preprint arXiv:1903.11406 (2019)
23. Trouillon T, Welbl J, Riedel S, Gaussier E, Bouchard G (2016) Complex embeddings for simple link prediction. In: Proceedings of the 33rd international conference on machine learning, pp 2071-2080
24. Wang Q, Mao Z, Wang B, Guo L (2017) Knowledge graph embedding: a survey of approaches and applications. TKDE 29, 12, pp 2724–2743
25. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the 28th AAAI conference on artificial intelligence, pp 1112–1119
26. Wawrzinek J, Pinto JMG, Wiehr O et al (2020) Exploiting latent semantic subspaces to derive associations for specific pharmaceutical semantics. Data Sci. Eng. 5(4):333–345
27. Wen J, Li J, Mao Y, Chen S, Zhang R (2016) On the representation and embedding of knowledge bases beyond binary relations. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 1300–1307
28. Wu S, Zhang Y, Gao C et al (2020) GARG: anonymous recommendation of point-of-interest in mobile networks by graph convolution network. Data Sci Eng 5(4):433–447
29. Xiao H, Huang M, Zhu X (2016) TransG: a generative model for knowledge graph embedding. In: Proceedings of the 54th annual meeting of the association for computational linguistics, pp 2316–2325
30. Yu L, Quanming Y, Yong L (2020) Generalizing tensor decomposition for n-ary relational knowledge bases. In: Proceedings of the web conference 2020, pp 1104–1114
31. Zhang Z, Cai J, Zhang Y, Wang J (2020) Learning hierarchy-aware knowledge graph embeddings for link prediction. In: Proceedings of the AAAI conference on artificial intelligence 34(03):3065–3072
32. Zhang R, Li J, Mei J, Mao Y (2018) Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In: Proceedings of the 2018 World Wide Web conference, pp 1185–1194
33. Zhang F, Wang X, Li Z, Li J (2021) Transrhs: a representation learning method for knowledge graphs with relation hierarchical structure. In: Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence, pp. 2987–2993
34. Zhang Y, Yao Q, Dai W, Chen L (2020) AutoSF: searching scoring functions for knowledge graph embedding. In: 2020 IEEE 36th international conference on data engineering (ICDE). IEEE, pp 433–444
35. Zhu M, Shen D, Xu L et al (2021) Scalable multi-grained cross-modal similarity query with interpretability. Data Sci Eng 6(3):280–293