Information Sciences 666 (2024) 120438

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multi-perspective knowledge graph completion with global and interaction features

Duantengchuan Li [a], Fobo Shi [b,c], Xiaoguang Wang [c], Chao Zheng [a], Yuefeng Cai [d], Bing Li [a,e,*]

[a] School of Computer Science, Wuhan University, Wuhan 430072, China
[b] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China
[c] School of Information Management, Wuhan University, Wuhan 430072, China
[d] ZTE Corporation, Shenzhen, 518057, China
[e] Hubei Luojia Laboratory, Wuhan 430079, China

ARTICLE INFO

Keywords:
Knowledge graph embedding
Global feature
Link prediction
Complex relations
Convolutional network
Interaction feature

ABSTRACT

Knowledge graphs are multi-relation heterogeneous graphs. Thus, the existence of numerous multi-relation entities imposes a tough challenge to the modelling of the knowledge graph. Some recent works represent the property of corresponding entities and relations by generating embeddings. They attempted to identify the missing entities by translation operations or semantic matching. However, the expressiveness of these approaches depends on the entity (relations) embedding. The heterogeneity of entities leads to the difficulty of balancing uniform embedding dimension settings on complex and sparse relational entities, as high-dimensional embedding leads to the overfitting of sparse relational entities, and low-dimensional embedding leads to the underfitting of complex relational entities. We introduce a multi-perspective knowledge graph embedding model with global and interaction features (MGIF) to alleviate these issues. This achieved knowledge transfer from complex relational entities to sparse relational entities through the multi-view features. In particular, to overcome the local limitations of convolution neural networks, the global features shared between entities (relations) and entities (relations) are incorporated in the MGIF. The performance of MGIF is experimentally evaluated on several datasets. The experimental effects demonstrate that MGIF can efficiently model complicated entities and accomplish state-of-the-art complex relationship prediction results on most evaluation metrics.

1. Introduction

A knowledge graph (KG) is a knowledge base of a semantic network consisting of nodes, edges and attributes. Recently, technologies associated with the KGs domain have been rapidly developed and widely used for other downstream tasks, including question answering [1,2], intelligent education [3], and recommendation systems [4,5]. According to the resource description framework (RDF) definition, triples are used as a general representation of KGs [6]. The KG contains a vast number of triples, represented in the form of (*head entity, relation, tail entity*), i.e. (h, r, t).

[*] Corresponding author at: School of Computer Science, Wuhan University, Wuhan 430072, China.
E-mail addresses: dtclee1222@whu.edu.cn (D. Li), foboshi99@gmail.com (F. Shi), wxguang@whu.edu.cn (X. Wang), bingli@whu.edu.cn (B. Li).

https://doi.org/10.1016/j.ins.2024.120438
Received 23 October 2023; Received in revised form 25 December 2023; Accepted 3 March 2024

Available online 7 March 2024

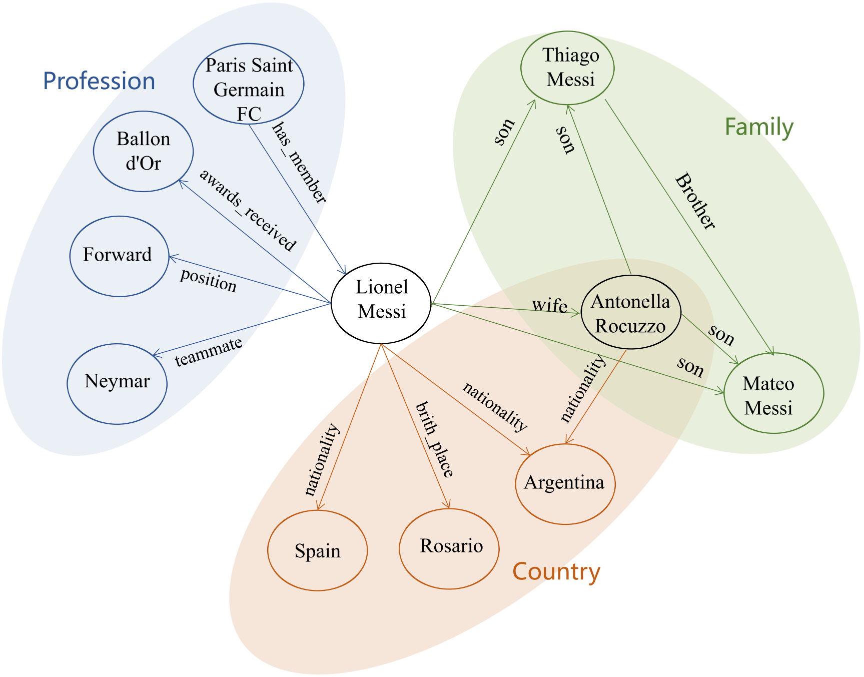


Fig. 1. An example of *Lionel Messi*'s multi-relation heterogeneous knowledge graph. In multiple perspectives, *Lionel Messi* plays different roles in different domains.

The KG is fundamentally a heterogeneous graph with multiple relations. The relations and entities in the graph correspond to the edges and nodes in the KGs. They are usually represented as low-dimensional vectors to distinguish between different nodes effectively. These vectors are called knowledge graph embeddings (KGE). The knowledge graph embedding model is mainly divided into *geometry-based Models* [7–11] and *semantic matching models* [12–16]. These methods are easy and efficient. However, due to the limitation of the number of parameters, these shallow models may be incapable of learning embedding with sufficient expressiveness. For these shallow methods, the embedding dimension is forced to increase to boost the expressiveness, which makes it tough to apply shallow models to large-scale KGs [17].

However, representing a diversity of semantic levels of entities in the multi-relational graph is the main challenge faced by real-world KGs, which may be what limits the performance of these neural network methods. Existing KGs intrinsically suffer from the heterogeneity of graphs when integrating multi-relational knowledge graphs. Many entities are connected to various types of relations, and directed edges generally represent relations. For example, Fig. 1 shows the entity *Lionel Messi* is attached to neighbouring nodes in different domains by multiple types of edges. The neighbour domains of *Lionel Messi* include three categories: “Profession”, “Family”, and “Country”. Intuitively, the representation of *Lionel Messi* is expected to be more oriented to reflect the semantic domain associated with the neighbourhood in the same triple. Specifically, in the triple (*Lionel Messi, position, Forward*), the embedding of *Lionel Messi* interacts with the relation *position*, and the entity *Forward* should be more relevant to the domain of “Profession”. In contrast, in the triple (*Lionel Messi, son, Mateo Messi*), the semantic representation produced when the entity *Lionel Messi* interacts with the relation *son* and the entity *Mateo Messi* is more oriented to represent the “Family” domain of *Lionel Messi*. We argue that most existing neural network approaches ignore the interaction specificity between nodes and edges. As a result, the obtained node (edge) embedding representation is not expressive or powerful enough to encompass its multi-aspect semantics [17,18].

Therefore, we present an advanced multi-perspective knowledge graph embedding model called MGIF, which incorporates global and interaction features. First, the adaptive relation-aware convolution module is designed to extract shared information between relations through relation-aware convolution kernels. Then, the convolutional interaction between entities and interaction-aware convolution kernels realizes knowledge transfer from relations to entities. Meanwhile, to capture the global knowledge, nodes and edges were fed into the global-aware convolution module to extract global features. Furthermore, the feature maps of the interaction-aware convolution and the global-aware convolution are integrated. Finally, the fused tensor is mapped to the embedding dimension, and the triple score is obtained by doing the inner product with the entity embedding matrix. The experimental results of MGIF demonstrate that MGIF can effectively tackle knowledge transfer between entities and relations and can simultaneously consider both the local interaction features and the global features of the entities on the KG.

1.1. Research objectives and contributions

Multi-relational entities in KG do not capture sufficiently generalized embedding representations from the training phase. Therefore, we propose incorporating global and interactive features to obtain a larger quantity and higher quality. In addition, the distribution of most entities (relations) in finding missing entities in KG needs to be more balanced, and we aim to fuse feature knowledge from multiple perspectives to enhance the expressiveness of the model. Leveraging the analysis mentioned above, we propose a KGE model incorporating global and interaction awareness, which fuses multi-view features to obtain richer semantic information. Our work has two overarching objectives: 1) To obtain a larger quantity and higher quality of interaction features between entities and relations to represent richer semantic structural information. 2) In the feature fusion phase, global information is incorporated to obtain entity (relation) embeddings with multi-perspective knowledge. The following are the main contributions of this article:

- A novel model MGIF is proposed to construct the multi-perspective features in knowledge graphs. MGIF generates three convolution filters from different aspects. The self-attention operator is employed to merge the three levels of features in the knowledge graphs and achieves multi-view knowledge awareness.
- To acquire the global characteristic of entities, the global-aware convolution kernels are used to convolve relations and entities separately. In addition, aiming to boost the quantity and quality of entity-relation interactions, we propose a data augmentation method and adaptive relation-aware convolution.
- After extensive and rich experiments on commonly used datasets, MGIF was observed to achieve several superior results on the evaluation metrics of the KGE task, especially in complex relations modelling experiments.

The rest of this work is structured as follows. Section 2 provides a relevant introduction and analysis of KGE. Section 3 explains the overall framework, model calculation details, and algorithm principles of the Multi-Graph Integration Framework (MGIF). Section 4 presents the experimental analysis based on experiments conducted on five publicly accessible datasets. Section 5 further discusses the theory and applications. Finally, conclusions and future work recommendations are discussed in Section 6.

2. Related work

The proposed KGE models can be broadly categorized into four families [19]: (1) *Geometry-based Models*, (2) *Semantic Matching Models*, (3) *Graph Network Models*, and (4) *Convolutional Network Models*.

2.1. Geometry-based models

Geometry-based KGE models typically model mapping relationships between head and tail entities as a geometric operation such as translation, scaling and rotation. This enables the ground truth to be found by projecting the head entity to the vector space of the tail entity in a relationship-specific mapping. TransE [7] was the first proposed translation-based model, which assumes that the entity and relation vectors satisfy $\mathbf{h} + \mathbf{r} = \mathbf{t}$. Equation (1) is the scoring function for the TransE:

$$\psi(h,r,t) = -d_r(h,t) = -\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{L_1/L_2}, \tag{1}$$

where $d_r(h,t)$ represents the calculated L_1 distance or L_2 distance of $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$. However, such a single spatial translation leads to all approximate tail entity embeddings obtained from complex relations converging to the same vector. To address this issue, TransH [10] projects the normal vectors generated by the entities through the relations into different hyperplanes. TransR [20] argues that an entity has multi-aspect characteristics and maps entities into different relational semantic spaces. The translational family of models is intended to deal with complex relations (1-N, N-1 and N-N) but ignores different kinds of relational patterns. To simulate the relational mapping of different types of relational patterns, RotatE [11] replaces the geometric operation from the translation with rotation, which can express models for various relational patterns, including symmetry, asymmetry, inversion and composition. After that, Rot-Pro [21] combines the two geometric operations of translation and rotation to model the transitivity pattern successfully. The advanced approach NFE [22] removes the traditional limitation of representation space and introduces uncertainty into the knowledge graph from a group-theoretic point of view, using the operator in the group as a geometric operation for relational mapping. However, recent work [23] has found through extensive experiments that all of these methods [10,11,21] only theoretically demonstrate that their geometric operations can infer multiple relational patterns in KG, yet do not perform well in prediction experiments of different relational patterns.

2.2. Semantic matching models

Semantic matching models are also known as bilinear or tensor decomposition models. The earliest and representative model of the semantic matching model is the RESCAL [13] model, which models the latent semantic pairwise interactions of entities by relation matrices. Specifically, the scoring function defined by RESCAL [13] is as follows:

$$\psi(h,r,t) = \mathbf{h}^{\top}\mathbf{M}_r\mathbf{t} = \sum_{i=0}^{d-1}\sum_{j=0}^{d-1}\left[\mathbf{M}_r\right]_{ij}\cdot[\mathbf{h}]_i\cdot[\mathbf{t}]_j, \tag{2}$$

where $\mathbf{h},\mathbf{t} \in \mathbb{R}^d, \quad \mathbf{M}_r \in \mathbb{R}^{d\times d}$, denotes multiplication operation, $\mathbf{M}_r$ is relation matrix. As the KGs increase, it leads to a more complex calculation of RESCAL. To alleviate this problem, DistMult [12] model constraints $\mathbf{M}_r = diag(\mathbf{r})$, where $\mathbf{r}$ is the semantic vector, $\mathbf{r}$ representing the relation and $diag()$ denotes the diagonal matrix consisting of vectors. To explore more appropriate representation spaces, ComplEx [15] embeds entities and relations into complex vector spaces, modelling the relational projection operations of entities by low-rank decomposition. TuckER [16] adopts a more advanced version of low-rank decomposition, the tucker decomposition, and proves that the DistMult [12] and ComplEx [15] can be considered as special cases of TuckER. From the perspective of reducing the size of the model, B-CP [24] proposes a binarized CANDECOMP/PARAFAC (CP) decomposition, which enables the model to compute semantic matching scores with larger embedding dimensions quickly. Generally, semantic matching models aim to enhance the model's capability. However, all of them have a challenge that requires a solution: as the size of the KG increases, the models all have to improve the embedding dimension, which in turn tends to lead to overfitting.

2.3. Graph network models

Graph Neural Networks (GNNs) are effective for handling graph-level tasks. Due to the topological similarity between GNN and KG, it is natural to utilize GNN for KGE tasks. R-GCN [25] is the first to consider relational information in modelling GNNs and to design aggregation functions containing relational information to suit the knowledge graph embedding task. To tackle the over-parameterization restrictions of the learning node representation in R-GCN, CompGCN [26] designs a variety of information aggregation functions and implements a unified framework that generalizes multiple GNN-based modelling. The representation space of traditional GCN is typically a real vector space, then ComplexGCN [27] follows [15] to replace the representation space and attempts to extend the embedding representation of nodes (edges) in GCN to complex vector space, using the hermitian inner product to construct the aggregation function. TE-ConvE [28] has made some counter-intuitive findings through extensive experiments: the process of capturing graph structure in the GCN-based KGE models [25,29,26] does not have a significant impact on performance. Instead, the transformation of the entity representation is the key reason for improved performance. To deal with local heterogeneity in the neighbours of an entity in KG, SHGnet [30] proposes a hierarchical aggregation architecture to mitigate the noise of heterogeneity on the entity representation. The GNN-based approaches are promising. However, most graph neural network models are computed over the whole graph and are not scalable on large-scale KGs [31]. They require huge computational resources to implement information transfer between nodes [28].

2.4. Convolutional network models

There is also interesting work on convolutional neural networks (CNN) available for KGE tasks. ConvE [17] is the first to adopt convolution neural networks to link prediction tasks. ConvE defines the following scoring function:

$$\psi(h,r,t) = f(\mathrm{vec}(g([\overline{\mathbf{h}};\overline{\mathbf{r}}] * \mathbf{K}))\mathbf{W})\mathbf{t}, \tag{3}$$

where $\mathrm{vec}(\cdot)$ represents vectorized operation. $\overline{\mathbf{h}}, \overline{\mathbf{r}}$ denote the head and tail entity after reshaping, respectively. $[\overline{\mathbf{h}};\overline{\mathbf{r}}]$ denotes stitch operation, $*$ denotes convolutional operation, $\mathbf{K}$ represents the convolution kernel, and $\mathbf{W}$ is the parameter of the fully connected projection layer. To enhance the interaction between entities and relations, InteractE [18] applied a chessboard-style reshape approach to obtain full interaction between entities. HypER [32] utilizes a hyper-network model to enable cross-layer weight sharing and dynamically synthesize weights based on inputs, thereby enabling adaptive entity-relation feature interactions. The method in ConvR [33] proposed to directly utilize relation embeddings as convolution kernels for convolutional operations on entities to substantially boost the number of interactions. Following [15,27], CirlularE [34] performs circular convolution of entity-relation interactions in the complex domain in an attempt to obtain a more expressive knowledge representation. SDFormer [35] and MSHE [36] introduce transformer and multi-hierarchical convolution, respectively, to further enhance the entity-relation interaction.

In contrast to GNN-based methods, CNN-based methods can be easily scaled to large-scale KGs with lower resource consumption. However, the aforementioned CNN-based method emphasizes the quantity of entity-relation interaction and neglects the quality of entity-relation interaction. For more qualified entity-relation interaction, we designed the adaptive relation-aware module to obtain more meaningful relations to entities' knowledge interaction. Inspired by InteractE [18], we applied a data augmentation approach that exponentially increases the number of entity-relation interactions with almost no additional parameters. Meanwhile, to overcome the local limitation of the CNN-based KGE model, we incorporate the global features to achieve entity-to-entity and relation-to-relation knowledge transfer, respectively. It is capable of mitigating the issue of information silos, which are created by isolated nodes lacking neighbours in the inner or outer degrees.

3. Methodology

This part will explore a method for incorporating global-aware and interaction-aware information. The framework of MGIF is shown in Fig. 2. In Section 3.1, the symbology used in the article is first introduced. Then, in Section 3.2, the MGIF is thoroughly detailed. The training objectives and optimization approach are drawn in Sections 3.3.

3.1. Problem formulation

The symbology of this chapter will be first introduced. The knowledge graph is represented by $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E} = \left\{e_1, e_2, \cdots, e_{|\mathcal{E}|}\right\}$ and $\mathcal{R} = \left\{r_1, r_2, \cdots, r_{|\mathcal{R}|}\right\}$ are the set of all relations and entities in the KG respectively. $|\mathcal{E}|$ represents the entity count, and $|\mathcal{R}|$ denotes the number of relations. $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the set of all triples in the Knowledge graph and consists of a set of triples (h, r, t). The bold symbols $\mathbf{h}, \mathbf{r}$, and $\mathbf{t}$ denote the vector representations of triples (h, r, t). Table 1 presents this article's primary notations and explanations.

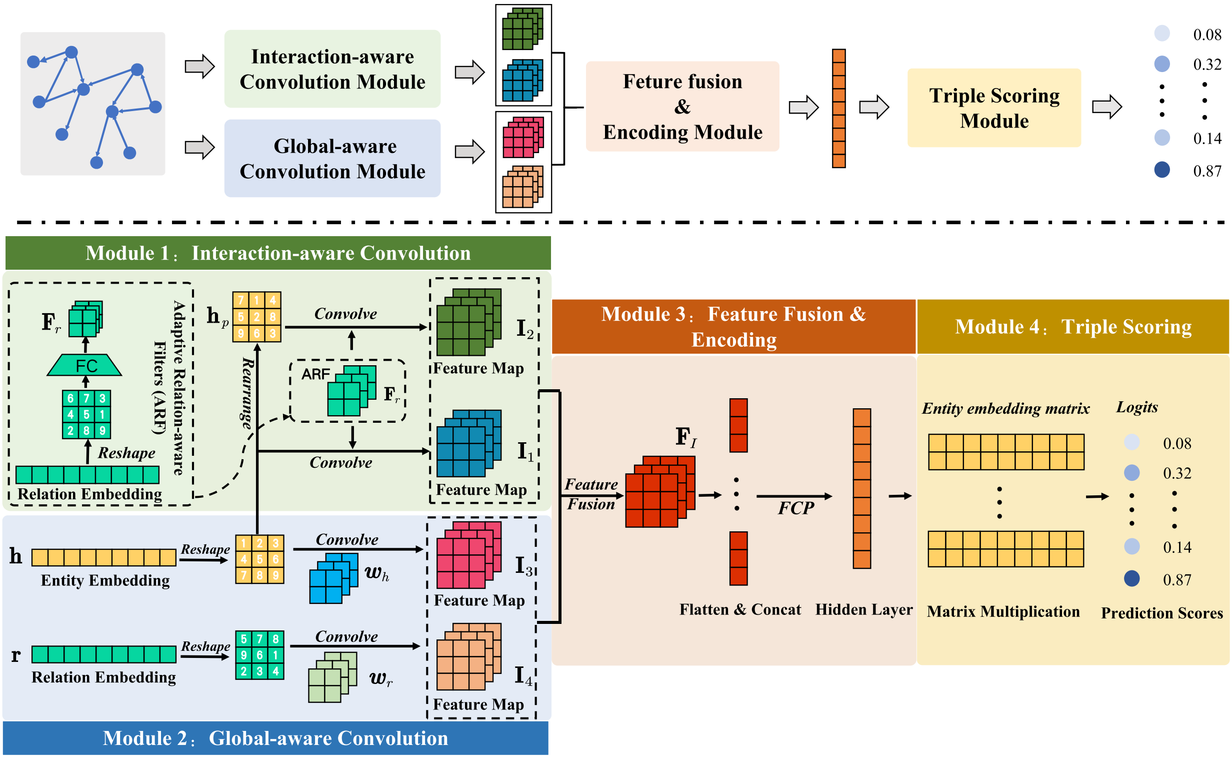


Fig. 2. Framework of the multi-perspective knowledge graph completion model with global and interaction features. It consists of four components: the interaction-aware convolution, the global-aware convolution, the features fusion & encoding and the triple scoring.

Table 1
Symbols and Descriptions.

Symbols	Descriptions
$\mathcal{G}$	Knowledge graphs
$\mathcal{E}, \mathcal{R}, \mathcal{T}$	Set of entities, relations, and triples
h, r, t	Head entity, relation and tail entity
$\mathbf{h}, \mathbf{r}, \mathbf{t}$	Embedding of head entity, relation and tail entity
$\psi(x)$	Score function
$\lvert\mathcal{E}\rvert, \lvert\mathcal{R}\rvert$	Number of entity, relation
$\mathbf{E}, \mathbf{R}$	Embedding matrices of entities and relation
N	Number of negative sampling
$\hat{y}(x)$	Triple prediction probability

3.2. The architecture of MGIF

A brief description of the framework of MGIF is shown in Fig. 2. MGIF is summed up as four modules: interaction-aware convolution, global-aware convolution, encoding representation, and triple scoring. The interaction-aware convolution module aims to encode relation-aware entity features. Such entity features are relation-specific and capture the local properties between relations and entities. In the global-aware convolution, the feature maps consider the global features of relations and entities separately. Afterwards, the feature maps obtained from global-aware and interaction-aware are fused to generate the corresponding encoding representation. Finally, the triple-scoring module picks up the entity obtained with the highest likelihood.

3.2.1. Interaction-aware convolution

Adaptive Relation-aware Filters Generation. Due to the relations being incompletely independent of each other in the KG, extracting interaction features between entity and relation are essential while subjecting all relations to some constraints, so the operation of sharing partial information is adopted. Specifically, a fully connected network that leverages shared parameters for the relationship is expanded by feature reorganization, and some of the same semantic information, or components, can exist in the result obtained. The network generates regular and constrained convolution kernels that allow meaningful interaction information for subsequent convolution with the head entity embedding. For a given KG, all entities and relations are randomly initialized into d-dimensional semantic feature embeddings to obtain the feature matrix $\mathbf{E}, \mathbf{R} \in \mathbb{R}^{|\mathcal{E}| \times d}$, in the prediction of triple (h, r, t), first, index $\mathbf{E}$ and $\mathbf{R}$ to get the head and tail entity embedding $\mathbf{h}, \mathbf{r} \in \mathbb{R}^d$. Then, the adaptive relation-aware convolution kernels of $\mathbf{r}$ are generated, noting them as $\mathbf{F}_r$ as:

$$\mathbf{F}_r = \phi\left(\mathbf{W}_1\mathbf{r} + \mathbf{b}_1\right), \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{ck_wk_h\times d}$, $\mathbf{b}_1 \in \mathbb{R}^{c\times k_w\times k_h\times d}$ represent the weight bias parameters that can be extracted for the fully connected layer, respectively, c, k_w, k_h denotes the number, length and width of the convolution kernels, respectively. ϕ is a reshape operation that rearranges the output of the fully connected network into the $c \times k_w \times k_h$ shape. After recomposing, it can be used as the convolution kernel for the head entity embedding in subsequent convolution operations.

Interaction-aware Features Extraction. This step will utilize the convolution kernel obtained in the previous step, and the following calculation is conducted next:

$$\mathbf{I}_1 = \mathbf{h} * \mathbf{F}_r, \tag{5}$$

where $*$ represents convolutional operations, $\mathbf{h} \in \mathbf{R}^{d_h\times d_w}$ denotes the head entity, and $d_h \times d_w = d$, $\mathbf{F}_r \in \mathbb{R}^{c\times k_w\times k_h}$, $\mathbf{I}_1 \in \mathbb{R}^{c\times m_w\times m_h}$, c denotes the number of feature map channels, m_w is the output feature map length, m_h denotes the width of output feature map.

The fully connected layer in the adaptive relation-aware convolution ensures shared information among all relational embeddings. This information ensures that follow-up interactions of relation and entity are meaningful, which reflects the interaction specificity of relation-aware and ensures the effectiveness of predictions. Meanwhile, using a relation-aware convolution kernel to interact with entities can capture more profound interaction specificity than directly applying relation embeddings as convolution kernels, which will theoretically speed up the convergence of the model and thus produce more effective interactions.

Data Augmentation. To have more interaction features between entities and relations with as few parameters as possible while improving the generalization capability of the model, the rearrangements obtained above are similarly applied to the convolution operation as:

$$\mathbf{I}_2 = \varrho(\mathbf{h}) * \mathbf{F}_r. \tag{6}$$

Where $\varrho()$ function is a rearrangement operation if a 10-dimensional embedding is input. Then the output is a random permutation of these 10-dimensional embedding, e.g. $[a, b, c, d, e, f, g, h, i, k] \rightarrow [e, a, f, g, b, i, c, h, k, j]$.

The embeddings obtained by rearrangement are convolved by the above data augmentation method. This data augmentation method can further boost the multi-layer and rich interactions between entities and relations without increasing parameters, effectively augmenting the number of interactions. Although the data augmentation module introduces additional parameters to MGIF, the number of parameters is acceptable and does not affect the extension of MGIF to larger-scale KGs. The specific number of parameters is analyzed in the subsequent experimental section.

To sum up, the above step obtains entity embeddings with relational specificity through the interaction-aware convolution. If the entity or relation changes, the interaction will differ due to the convolutional network's input and relation-aware convolution kernel change. Therefore, this interaction result is unique to the corresponding relations and entities, and the feature maps with interaction-specific are generated by interaction-aware convolution.

3.2.2. Global-aware convolution

This long-tail distribution also tends to exist for entities and relations in KG. The entity that rarely emerges in the triples is probably forming data silos. It is a challenge to encode their interactive features. Therefore, we design a global-aware convolution to extract global information about various relations and entities, respectively. Global information represents that the information and constituents presented are the same regardless of whether the relevant entities and relations interact (i.e., relations and entities) change.

Global Entity-share Convolution. In the KG, although most nodes appear in the triples with low frequency, there are still a few entities with high frequency. In addition, the number of interactions of these high-frequency entities with relations is much more than that of the low-frequency entities. Inspired by this, we consider using high-frequency entities to help low-frequency entities obtain more meaningful embeddings. Specifically, convolving all entities after reshaping with global entity-share filters makes low-frequency entities and high-frequency entities share the same convolution kernel. The interaction information of high-frequency entities can be transferred to low-frequency entities in a global knowledge-sharing manner. This is done as follows:

$$\mathbf{I}_3 = \mathbf{h} * \mathbf{K}_h. \tag{7}$$

Global Relation-share Convolution. This part is similar to the previous step and will not be repeated. The relation-share convolution process is as follows:

$$\mathbf{I}_4 = \mathbf{r} * \mathbf{K}_r, \tag{8}$$

where $\mathbf{K}_h, \mathbf{K}_r \in \mathbb{R}^{c\times k_w\times k_h}$, $*$ denotes the convolution operation. Compared with the previous models, the past models are mostly convolutional after splicing $\mathbf{h}, \mathbf{r}$. The emphasis is on the interaction between relations and entities, thus ignoring the fact that some generalized global information is required between relation and entity embeddings. The feature maps $\mathbf{I}_3$, $\mathbf{I}_4$ denote obtained by global entity-share convolution and global relation-share convolution, respectively.

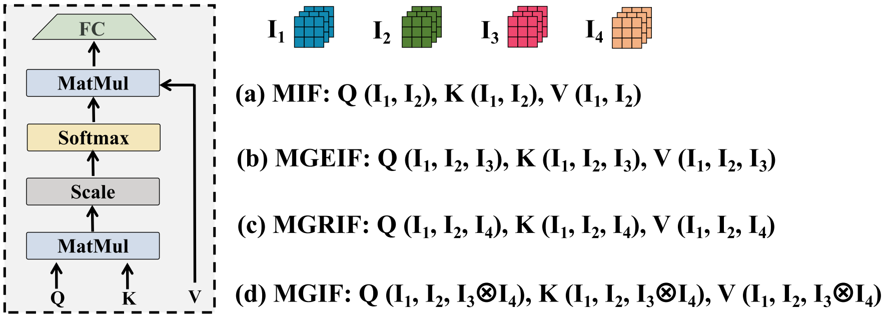

FC
MatMul
Softmax
Scale
MatMul
Q
K
V
I1
I2
I3
I4
(a) MIF: Q (I1, I2), K (I1, I2), V (I1, I2)
(b) MGEIF: Q (I1, I2, I3), K (I1, I2, I3), V (I1, I2, I3)
(c) MGRIF: Q (I1, I2, I4), K (I1, I2, I4), V (I1, I2, I4)
(d) MGIF: Q (I1, I2, I3⊗I4), K (I1, I2, I3⊗I4), V (I1, I2, I3⊗I4)

Fig. 3. Comparison of MGIF and three variants of multi-perspective feature map fusion process: (a) the Multi-perspective Interaction Feature model (MIF) uses feature maps $\mathbf{I}_1$ and $\mathbf{I}_2$ in the feature fusion process. (b) The multi-perspective Global Entity and Interaction Feature model (MGEIF) adopts feature maps $\mathbf{I}_1$, $\mathbf{I}_2$ and $\mathbf{I}_3$ in the features fusion process. (c) The multi-perspective Global Relation and Interaction Feature model (MGRIF) adopts feature maps $\mathbf{I}_1$, $\mathbf{I}_2$ and $\mathbf{I}_4$ in the feature fusion process. (d) The multi-perspective Global and Interaction Feature model (MGIF) adopts feature maps $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$ and $\mathbf{I}_4$ in the features fusion process. ⊗ denotes the composition between global-aware feature map $\mathbf{I}_3$ and $\mathbf{I}_4$, such as addition, multiplication, etc.

3.2.3. Features fusion and encoding

The above process can extract global features and relation interaction features in regular prediction, reduce overfitting problems, improve generalization ability, and also achieve better prediction when dealing with sparse relations or entities. The next step is to fuse the yielded feature maps with global-aware and interaction-aware. We propose MGIF and its three variants based on different approaches to feature fusion.

As shown in Fig. 3(a), the Multi-perspective Interaction Feature model (MIF) uses feature maps $\mathbf{I}_1$ and $\mathbf{I}_2$ in the feature fusion process. That is,

$$\mathbf{F}_I = f\left(attn_{self}\left(\mathrm{I}_1 \| \mathrm{I}_2\right)\right)\mathbf{W}_2 + b_2. \tag{9}$$

In Fig. 3(b), the feature maps $\mathbf{I}_1$, $\mathbf{I}_2$ and $\mathbf{I}_3$ are applied in the Multi-perspective Global Entity and Interaction Feature model (MGEIF). The feature fusion process can be written as:

$$\mathbf{F}_I = f\left(attn_{self}\left(\mathrm{I}_1 \| \mathrm{I}_2 \| \mathrm{I}_3\right)\right)\mathbf{W}_2 + b_2. \tag{10}$$

In Fig. 3(c), the Multi-perspective Global Relation and Interaction Feature model (MGRIF) adopts feature maps $\mathbf{I}_1$, $\mathbf{I}_2$ and $\mathbf{I}_4$ in the feature fusion process, which is defined as follows:

$$\mathbf{F}_I = f\left(attn_{self}\left(\mathrm{I}_1 \| \mathrm{I}_2 \| \mathrm{I}_4\right)\right)\mathbf{W}_2 + b_2. \tag{11}$$

In Fig. 3(d), the Multi-perspective Global and Interaction Feature model (MGIF) adopts feature maps $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$ and $\mathbf{I}_4$ in the feature fusion process. The formula for the feature fusion is,

$$\mathbf{F}_I = f\left(attn_{self}\left(\mathrm{I}_1 \| \mathrm{I}_2 \| \left(\mathrm{I}_3 \otimes \mathrm{I}_4\right)\right)\right)\mathbf{W}_2 + b_2, \tag{12}$$

where $\mathbf{W}_2 \in \mathbb{R}^{3(m_w\times m_h)\times(m_w\times m_h))}$, $\mathbf{I}_\star \in \mathbb{R}^{c\times(m_w\times m_h)}$, $\star$ can be $1,2,3,4$. $\|$ is a splicing operation. $\otimes$ represents an composition operation between I_3 and I_4 feature maps. $attn_{self}(\cdot)$ denotes the correction of multi-perspective feature maps adopting the self-attention approach, whereby we expect that feature maps from different perspectives will pay attention to the relation and entity features under other views. $f(\cdot)$ denotes the activation function ReLU, $\mathbf{b}_2 \in \mathbb{R}^{c\times(m_w\times m_h)}$. Simply put, the splicing operation splashes the output feature maps of the convolution operation. For example, after stitching, a 32×200 feature map will transform into 32×800. Then, after fusion through the self-attention layer downscaling to 32×200, the features fusion process is shown in Fig. 3. We have tried other fusion methods, such as summation, weighted splicing fusion, pooling fusion, etc., and found that the self-attention layer is more effective as it can fuse effectively with a global field of view.

3.2.4. Triple scoring

The above-fused results are vectorized and then mapped to the same semantic space by the entity embedding from a fully connected layer. Therefore, the scoring function of the MGIF model is defined as:

$$\psi(h,r,t) = \sigma\left(f\left(\mathrm{vec}\left(\mathbf{F}_I\right)\mathbf{W}_3 + \mathbf{b}_3\right)\mathbf{t}\right), \tag{13}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization process flattens the vector. For example, $\mathbf{F}_I \in \mathbb{R}^{c\times m_e\times m_n} \to \mathbf{F}_I \in \mathbb{R}^{\left(c\times m_e\times m_n\right)}$. $\mathbf{W}_3, \mathbf{b}_3$ are the fully connected layer's parameters. The spreading result of the output feature map is mapped to the entity dimension by the fully connected layer. $\sigma(\cdot)$ and $f(\cdot)$ represent the sigmoid and ReLU activation function.

Algorithm 1: MGIF algorithm.

```
Data: Knowledge graph $\mathcal{G}=\{\mathcal{E},\mathcal{R},\mathcal{T}\}$.
Input: Training set $\mathcal{S}=(h,r,t)$;
       $ent.$ sets $\mathcal{E}$ and $rel.$ sets $\mathcal{R}$;
       batch size $b$ and learning rate $lr$;
       score function $\psi(h,r,t)$;
       early stop parameter $es$;
       negative sample $n$.
Output: $ent.$ embedding matrix $\mathbf{E}$;
        $rel.$ embedding matrix $\mathbf{R}$.
Data pre-processing;
Extracting the first set of randomly sampled triples $\mathcal{T}_{batch}$;
Random initialization of the embedding matrix $\mathcal{E}$ and $\mathcal{R}$ as $\mathbf{E}\in\mathbb{R}^{|\mathcal{E}|\times d}$ and $\mathbf{R}\in\mathbb{R}^{|\mathcal{R}|\times d}$;
loop
    sample($\mathcal{T},b$);  // extracting a set of randomly sampled triplets $\mathcal{T}_{batch}$
    for $(h,r,t)\in\mathcal{T}_{batch}$ do
        for $t'\in\mathcal{E}$ do
            $\mathcal{T}'_{batch}:(h,r,t')\leftarrow(h,r,t)$,  // sample corrupted triplet
            $\mathcal{T}'_{batch}=\{(h,r,t_1),(h,r,t_2),\ldots,(h,r,t_n)\}$, where $\{(h,r,t_1),(h,r,t_2),\ldots,(h,r,t_n)\}\notin\mathcal{G}$
        $\hat{y}(h,r,t)\leftarrow\psi(h,r,t)$  // Return the correct triple prediction probability.
        $\hat{y}(h,r,t')\leftarrow\psi(h,r,t')$  // Return the corrupted triple prediction probability.
    update embedding matrix $\mathbf{E},\mathbf{R}$.
end loop;
```

3.3. Training and optimization

For training MGIF, the optimizer uses a cross-entropy loss function with label smoothing for optimization; the loss function is as follows:

$$\mathcal{L}=-\frac{1}{N}\sum_{i=1}^{N}[y\log(\psi(h,r,t))+(1-y)\log(1-\psi(h,r,t))]. \tag{14}$$

If (h,r,t) is a true triplet, then $y=1$, otherwise $y=0$. Following processing head entities and relations during training, we calculate scores for N tail entities. All triples are represented throughout this operation as $(h,r,\triangle)$, where $\triangle$ corresponds to the tail entity. The training objective is to enable the score $\psi(h,r,t)$ of the true triple to be as close to 1 as possible, otherwise 0. The decreasing loss indicates the degree of completion of this objective. For the input (h,r), a list of size N is generated, corresponding to different entities, indicating that there is a triplet containing the above entity as the tail entity. True 1 will be shown in the list; otherwise, 0. y is the list in the above process, and $\psi(h,r,t)$ is the list calculated by the evaluation function. During the experiment, N represents the total count of entities in KG. $N=|\mathcal{E}|$ is called 1-N scoring. N can also be equal to $x+1$ for each group of entity-relation pair (h,r), where x are corrupted entities selected by negative sampling operation. Specifically, the negative sampling operation randomly chooses some entity to act as the tail entity of a triple. The process is as follows:

$$(h,r,t)\rightarrow(h,r,t'). \tag{15}$$

The detailed training and optimization process procedure of the MGIF is described in Algorithm 1. In detail, the model takes Xavier [37] random initialization for all embedding parameters, which can accelerate the model fitting speed and alleviate the gradient explosion or gradient disappearance problem [38]. Batch normalization [39] is separately applied after the convolution and feature mapping processes to enhance convergence speed. To prevent the network from overfitting, dropout [40] is adopted after the embedding input. The convolution operation's feature map output is mapped from the feature map to the tail entity space. Meanwhile, label smoothing [41] is utilized to strengthen the generalization capacity of MGIF while mitigating overfitting.

4. Experiments and analysis

4.1. Experiments setup

4.1.1. Datasets

In this section, we will select some datasets often adopted in related work in the direction of link prediction for relevant experiments. After screening and analysis, five datasets are selected, namely, FB15k,[1] FB15k-237,[2] WN18,[3] WN18RR,[4] and YAGO3-10.[5] Table 2 presents the statistical information of these datasets.

[1] https://everest.hds.utc.fr/lib/exe/fetch.php?media=en:fb15k.tgz.

[2] https://www.microsoft.com/en-us/download/details.aspx?id=52312.

[3] https://everest.hds.utc.fr/lib/exe/fetch.php?media=en:wordnet-mlj12.tar.gz.

[4] https://paperswithcode.com/dataset/wn18rr.

[5] https://github.com/ZhenfengLei/KGDatasets/tree/master/YAGO3-10.

Table 2
Knowledge Graph Datasets Statistics.

Datasets	*# Entities*	*# Relations*	*# Triples*		
			Train	*Valid*	*Test*
FB15k [7]	14,951	1,345	483,142	50,000	59,071
WN18 [7]	40,943	18	141,442	5,000	5,000
FB15k-237 [42]	14,541	237	272,115	17,535	20,466
WN18RR [17]	40,943	11	86,835	3,034	3,134
YAGO3-10 [17]	123,182	37	1,079,040	5,000	5,000

4.1.2. Evaluation metrics

Given a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, the model performance is evaluated by giving entity-relation pairs, feature encoding the entity-relation pairs to get the corresponding embeddings, and then computing with all candidate tail entities to derive a score for each candidate tail entity with a score value between 0 and 1. The score ranking of the entity is regarded as the prediction result of the model. Finally, the model performance is measured by ranking the obtained scores in terms of Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hitting Rate before k (Hits@k).

MR denotes that, after scoring the candidates, the rank of the true tail entity among the candidate tail entities is determined for a given triple. The complete test set is calculated using the same procedure as follows:

$$\text{MR}: \frac{1}{|\mathcal{T}_{test}|} \sum_{x_i \in \mathcal{T}_{test}} \left(rank_i \right), \tag{16}$$

where $\mathcal{T}_{test}$ represents the test set, $|\mathcal{T}_{test}|$ represents the number of test set triples. All the negative sampling operations are equivalent to replacing the tail entities, and the sorting operations are also sorting the tail entities. $rank_i$ denotes the position of the real tail entity within the potential tail entities, and any subsequent formula with the same representation represents the same meaning and will not be repeated.

MRR is a statistical indicator in MR replaced by the inverse. Finally, the sum of the inverse of each ranking is counted and divided by $|\mathcal{T}_{test}|$, and the calculation process is as follows:

$$\text{MRR}: \frac{1}{2\,|\mathcal{T}_{test}|} \sum_{x_i \in \mathcal{T}_{test}} \left(\frac{1}{\text{rank}_i^h} + \frac{1}{\text{rank}_i^t} \right). \tag{17}$$

The variable $rank_i^h$ denotes the position of the real head entity among the potential entities. $rank_i^t$ indicates the position of the correct tail entity among the potential tail entities.

Hits@k ranks the true tail entity of the triple in MR among the candidate tail entities; if it is less than or equal to, it is recorded as 1. Otherwise, it is recorded as 0. That is,

$$\text{Hits@}k: \frac{1}{|\mathcal{T}_{test}|} \sum_{x_i \in \mathcal{T}_{test}} f(rank_i), \tag{18}$$

$$f(x) = \begin{cases} 1, x \leqslant k \\ 0, x > k \end{cases}. \tag{19}$$

Among the metrics mentioned above, a lower MR value indicates improved overall model prediction performance, while a higher MRR value indicates improved overall model prediction performance. A higher value of Hits@k also signifies a more accurate model prediction.

4.1.3. Experimental settings

For the MGIF model, different combinations of hyperparameters are tried, and a grid search is used to acquire suitable configurations for the test and validation sets. The range of grid search hyperparameters was as follows: entity embedding and relation embedding dimension range $[10 \times 10, 20 \times 10, 15 \times 15, 20 \times 20]$, learning rate is in the range of $[0.001, 0.0001]$. The dropout rates of input, feature map, and hidden layer range $[0.1, 0.2, 0.3, 0.4, 0.5]$, batch size range $[128, 256]$, filter size range $[3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9]$, the number of convolution kernels range $[32, 64, 96]$.

The following parameters were applied to all datasets: learning rate 0.001, entity embedding and relation embedding dimension 15×15, and label smoothing 0.1. Specifically, the embedding dropout, feature map dropout, and hidden dropout were set to 0.5, 0.2, 0.5, with a batch size of 128, for the FB15k-237 dataset for the WN18RR dataset, embedding dropout, feature map dropout and hidden dropout were set to 0.2, 0.3, 0.3, with a batch size of 128, for the FB15k dataset, embedding dropout, feature map dropout, hidden dropout were set to 0.2, 0.2, 0.2, with a batch size of 256. For the WN18 dataset, the values of embedding dropout, feature map dropout, and hidden dropout were set to 0.2, 0.3, and 0.3, respectively. The batch size is set to 256. For the YAGO3-10 dataset, the input dropout, feature map dropout, and hidden dropout values were set to 0.2, 0.2, and 0.3, respectively. All experiments were conducted using a computer server equipped with a Tesla V100 GPU.

Table 3
Link prediction results on FB15k-237 and WN18RR. The optimal performance is indicated in **bold** and the second-best is underlined.

Model	FB15k-237					WN18RR				
	MR	MRR	Hits			MR	MRR	Hits		
			@1	@3	@10			@1	@3	@10
Translation-based models										
TransE [7]	225	0.288	0.192	0.325	0.478	5955	0.192	0.021	0.367	0.435
RotatE [11]	181	0.318	0.216	0.358	0.525	3335	0.474	0.425	0.49	0.571
PairRE [43]	-	-	-	-	-	-	0.454	0.411	0.469	0.548
Rot-Pro [21]	201	0.344	0.246	0.383	0.540	2815	0.457	0.397	0.482	**0.577**
NFE-1 [22]	-	0.355	0.261	-	0.543	-	**0.483**	0.438	-	0.576
Semantic Matching models										
DistMult [12]	407	0.180	0.101	0.191	0.347	4045	0.322	0.243	0.371	0.463
ComplEx [15]	548	0.233	0.145	0.266	0.408	4890	0.38	0.337	0.423	0.472
TuckER [16]	182	0.354	0.262	0.388	0.535	6684	0.463	0.435	0.478	0.516
B-CP [24]	-	0.295	0.210	0.324	0.483	-	0.480	**0.450**	0.490	0.530
GNN-based models										
R-GCN [25]	-	0.249	0.151	0.264	0.417	-	0.226	0.157	0.269	0.376
SACN [29]	-	0.350	0.260	0.390	0.540	-	0.470	0.430	0.480	0.540
CompGCN [26]	197	0.355	0.264	0.390	0.535	3533	0.479	0.443	0.494	0.546
ComplexGCN [27]	-	0.338	0.245	0.371	0.524	-	0.455	0.423	0.468	0.516
LTE-ConvE [28]	-	0.355	0.264	0.389	0.535	-	0.472	0.437	0.485	0.544
SAttLE [44]	-	0.358	0.266	0.394	0.541	-	0.476	0.442	0.490	0.540
SHGNet [30]	-	0.355	0.268	0.395	**0.544**	-	0.476	0.448	**0.496**	0.549
CNN-based models										
ConvE [17]	244	0.318	0.231	0.351	0.493	4187	0.430	0.400	0.440	0.520
HypER [32]	250	0.341	0.252	0.376	0.52	5798	0.465	0.436	0.477	0.522
ConvR [33]	-	0.350	0.261	0.385	0.528	-	0.475	0.443	0.489	0.537
InteractE [18]	181	0.355	0.263	0.390	0.539	5105	0.465	0.433	0.481	0.526
CirlularE [34]	**166**	0.347	0.252	0.385	0.538	**2605**	0.470	0.422	0.491	0.562
SDFormer [35]	185	0.356	0.264	0.390	0.541	3633	0.458	0.425	0.471	0.528
MSHE [36]	-	0.356	0.264	0.392	0.544	-	0.461	0.429	0.473	0.530
MIF(ours)	175	0.357	0.266	0.391	0.541	4304	0.470	0.433	0.485	0.543
MGRIF(ours)	175	0.360	0.269	0.395	0.543	4206	0.472	0.432	0.489	0.548
MGEIF(ours)	174	0.354	0.262	0.390	0.537	4342	0.471	0.430	0.487	0.550
MGIF(ours)	171	**0.362**	**0.271**	**0.398**	**0.544**	4198	0.475	0.433	0.494	0.557

4.2. Results and analysis

4.2.1. Evaluation on link prediction

In this part, to evaluate the effectiveness of the models, we used the datasets above, and the corresponding results and analysis will be offered. In the comparison table of prediction results, the optimal performance is indicated in **bold**, and the second-best is underlined.

Table 3 shows the results of the link prediction task. On FB15k-237, MGIF and its three variants perform very well compared to most baselines. Overall, MGIF improves the MRR value by 1.2% over the SAttLE [44], which is the best of the baseline methods. It also achieved optimal performance on Hits@1, Hits@3 and Hits@10 and sub-optimal performance on MR metrics. Specifically, compared with ConvE and HypER, which are also convolutional neural network-based models, MR is reduced by 28% and 29.6%, MRR is improved by 11.1% and 5.9%, Hits@1 is improved by 13.9% and 7.1%, Hits@3 is improved by 10.1% and 5%, and Hits@10 is enhanced by 8% and 4.2%, respectively.

In the experimental results of the WN18RR dataset, the performance of MGIF on this dataset is not spectacular, which may be related to the greater sparsity of WN18RR and the lack of entity-relation interactions of MGIF. However, MGIF achieves sub-optimal values for Hits@3, and its performance on MRR, Hits@1 and Hits@10 is not far from that of SOTA. Specifically, compared to the latest also based on the convolutional models SDFormer [35] and MSHE [36], the results of MGIF on MRR were improved by 3.2% and 3.0%, respectively, and in hits@10 The results on increased by 5.5% and 5.1% respectively.

In conclusion, the models of other approaches have performed well enough. However, they have yet to pay attention to the global information of relations and entities in the whole KG to some extent in the modelling process. However, MGIF, which incorporates global information, achieves similar or even outperforms the best models of the various other methods. Even when compared with advanced GNN-based models, which require massive computational resources and are limited to being extended to large-scale knowledge graphs, MGIF achieves comparable performance in most evaluation metrics.

FB15k and WN18 have been common datasets for link prediction tasks in the past and have been used less since the test set leakage problem was found. However, conducting experiments on these two datasets is also of some significance. The results on these two datasets are displayed in Table 4, showing that MGIF outperforms any previous models on the FB15K dataset. Specifically,

Table 4
Results of the Link Prediction by MRR and Hits@*k* on FB15k and WN18 Datasets.

Models	FB15k				WN18			
	MRR	Hits			MRR	Hits		
		@1	@3	@10		@1	@3	@10
TransE [7]	0.380	0.231	0.472	0.641	0.454	0.089	0.823	0.934
DistMult [12]	0.654	0.546	0.733	0.824	0.822	0.728	0.914	0.936
ComplEx [15]	0.692	0.599	0.759	0.840	0.941	0.936	0.936	0.947
R-GCN [25]	0.696	0.601	0.760	0.842	0.814	0.686	0.928	0.955
ConvE [17]	0.657	0.558	0.723	0.831	0.943	0.935	0.946	0.956
ToursE [8]	0.733	0.674	0.771	0.832	0.947	0.943	0.950	0.954
TuckER [16]	0.746	0.666	0.802	0.879	0.948	0.943	0.952	0.958
RotatE [11]	0.782	0.735	0.823	0.882	0.948	0.943	0.952	0.958
HypER [32]	0.790	0.734	0.829	0.885	**0.951**	**0.947**	**0.955**	0.958
B-CP [24]	0.733	0.660	0.793	0.870	0.945	0.941	0.948	0.956
CirlularE [24]	0.790	0.731	0.831	0.887	0.950	0.945	0.954	**0.960**
SDFormer [35]	0.692	0.628	0.732	0.802	0.948	0.944	0.951	0.957
MSHE [36]	-	-	-	-	0.948	0.943	0.951	0.952
MGIF(ours)	**0.805**	**0.752**	**0.843**	**0.893**	0.950	0.946	0.952	0.957

Table 5
Predicting Head and Tail Entity Results on FB15k-237 Dataset.

Model	Predict Head				Predict Tail			
	MRR	Hits			MRR	Hits		
		@1	@3	@10		@1	@3	@10
ConvE [17]	0.211	0.132	0.231	0.368	0.416	0.323	0.457	0.601
SACN [29]	0.241	0.158	0.260	0.409	0.446	0.352	0.490	0.631
RotatE [11]	0.239	0.149	0.265	0.424	0.432	0.329	0.477	0.639
InteractE [18]	0.258	0.170	0.283	0.437	0.454	0.358	0.498	0.644
SDFormer [35]	0.257	0.168	0.283	0.436	0.454	0.358	0.498	0.643
MIF(ours)	0.256	0.171	0.279	0.432	0.456	0.361	0.500	0.643
MGRIF(ours)	0.261	0.173	**0.286**	**0.440**	**0.459**	**0.364**	**0.504**	**0.646**
MGEIF(ours)	0.254	0.166	0.282	0.430	0.455	0.360	0.500	0.643
MGIF(ours)	**0.262**	**0.175**	**0.286**	0.438	**0.459**	**0.364**	0.503	**0.646**

Table 6
Prediction Results of Complex Relations by Hits@10 on FB15k Dataset.

Models	Predict Head Entity (Hits@10)				Predict Tail Entity (Hits@10)			
	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
TransE [7]	0.437	0.657	0.182	0.472	0.437	0.197	0.667	0.500
DistMult [12]	0.889	0.954	0.539	0.795	0.884	0.624	0.949	0.879
ComplEx [15]	0.885	0.947	0.404	0.683	0.888	0.511	0.939	0.750
ConvE [17]	0.866	0.959	0.598	0.786	0.852	0.704	0.957	0.858
TorusE [8]	0.775	0.906	0.479	0.859	0.773	0.592	0.893	0.890
RotatE [11]	**0.929**	0.967	0.602	**0.893**	0.923	0.713	0.961	**0.922**
HypER [32]	0.924	**0.968**	0.634	0.891	0.924	0.731	**0.962**	0.920
MGIF(ours)	0.925	**0.968**	**0.688**	0.891	**0.925**	**0.824**	0.959	0.921

the MRR is improved by 22.5% and 1.9%, Hits@1 by 34.8% and 2.5%, Hits@3 by 16.6% and 1.7%, and Hits@10 by 7.5% and 1.0%, respectively, when compared with the ConvE and HypER models, which are also CNN-based models. Furthermore, the results of MGIF are also superior compared to other models.

It is worth noting that MGIF does not achieve the same excellent performance on WN18 as on FB15k, and the same phenomenon occurs on WN18RR and FB15k-237. We analyze this as due to the type of knowledge graph. Because FB15k-237 and FB15k belong to the commonsense knowledge graph, which contains more complex relationships, MGIF with multi-view interaction capability can be fully leveraged. On the other hand, WN18RR and WN18 belong to the linguistic knowledge graph, which has relatively single structural levels, making MGIF's multi-perspective interaction ability not fully implemented. Thus, MGIF's advantage over the baseline model in WN18 and WN18RR is relatively small. However, in general, MGIF still achieves sub-optimal performance on all evaluation metrics on the WN18 dataset.

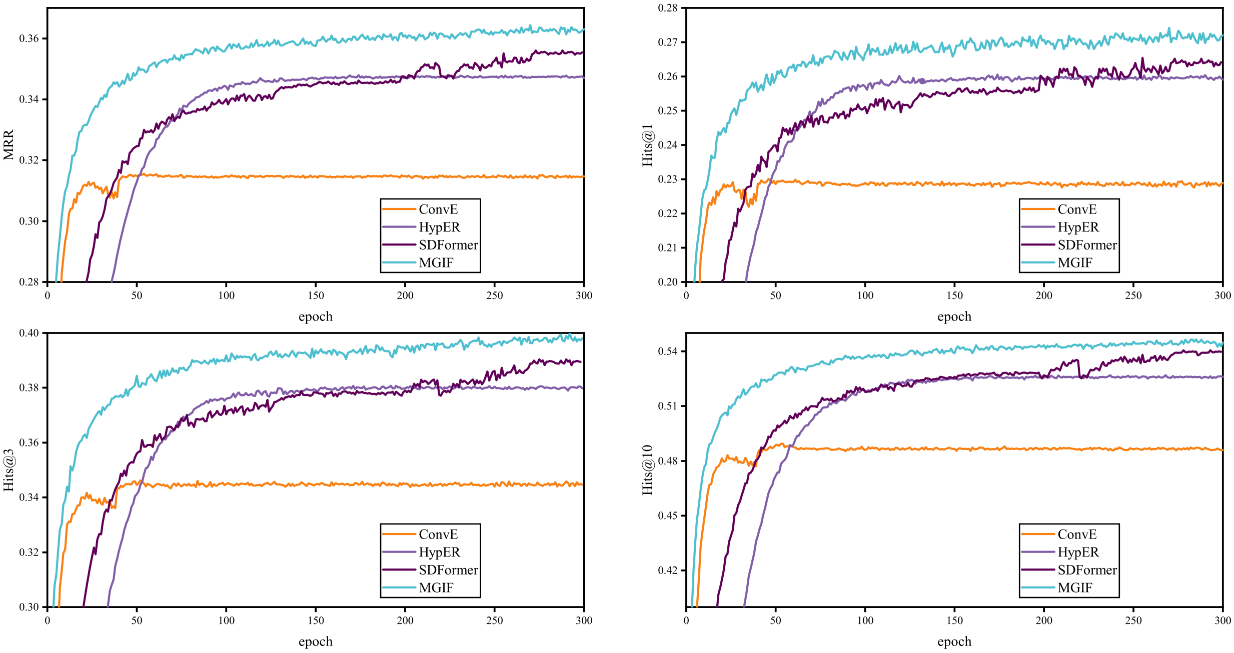


Fig. 4. Training processes compared with MRR, and Hits@1,3,10 on FB15k-237 datasets.

4.2.2. Complex relations modelling

The KG has a substantial number of complex relations. Whether these complex relations can be accurately predicted is critical to evaluating the model. To further evaluate the model, we compared the performance of MGIF with several baselines on the FB15k-237 dataset for predicting the absence of head and tail entities. Among them, InteractE is one of the excellent models based on CNN, and the obtained results are represented in Table 5, where those marked in black indicate the best values. However, it can be seen that MGIF again outperforms the compared baseline in all evaluation metrics.

According to TransE [7], the relations in the KG can be classified into four types easily, which are 1-to-1, 1-to-N, N-to-1, and N-to-N. For example, M-to-N means a certain relation exists with M alternatives for the head entity and N alternatives for the tail entity. Therefore, for each relation r, it is required to calculate the average number of head entities hpt for each tail entity and the average number of tail entities tph for each head entity in the triples it forms, and finally, it is divided into four types of relations according to the size of the values, as specified below:

$$\begin{cases} hpt < \eta \text{ and } tph < \eta \Rightarrow 1-\text{to}-1 \\ hpt < \eta \text{ and } tph \geq \eta \Rightarrow 1-\text{to}-\text{N} \\ hpt \geq \eta \text{ and } tph < \eta \Rightarrow \text{N}-\text{to}-1 \\ hpt \geq \eta \text{ and } tph \geq \eta \Rightarrow \text{N}-\text{to}-\text{N} \end{cases}, \tag{20}$$

where η is generally taken as 1.5 to ensure the fairness of the experiment.

The experiments were executed on the FB15k dataset, and Table 6 illustrates the prediction outcomes. MGIF has the best results for most of the metrics. Among all the prediction results, MGIF performs remarkably well in predicting the N-to-1 relations for head entities and the 1-to-N relations for tail entities, which are complex relationships. Specifically, the prediction results of MGIF for the N-to-1 relations in predicting head entities are compared with the same type of CNN-based models ConvE and HypER improved by 15.1% and 8.6%, respectively. The MGIF model outperforms the ConvE and HypER models by 17%, 12.7%, and 15.6%, respectively, in predicting the 1-to-N relation for tail entities.

From the above results, MGIF incorporating global information can tackle complex relations well and effectively alleviate the problem of data heterogeneity. Therefore, the multi-perspective features enable the model to capture information about the characteristics of many entities (relations) in different domains. The multi-perspective features provide the model with the knowledge of entities (relations) in multi-aspects to obtain entity (relations) embeddings conducive to modelling complex relations.

4.2.3. Convergence analysis

By comparing the experimental results of ConvE [17], HypER [45], SDFormer [35], and MGIF in the Fig. 4, which are all CNN-based models, we can observe that HypER is more complex than ConvE. Hence, the convergence of the model is relatively slow, i.e., about the first 50 epochs, HypER does not work as well as ConvE, but later on, it overtakes, which proves the effectiveness of HypER. However, the effect of MGIF has been better than ConvE, HypER, and SDFormer, which supports that MGIF is not only the optimal performance but also faster than other models regarding the convergence speed.

Table 7
Impact of Two Types of Global Features Compound on FB15k-237 Datasets.

Global feature composition type $\otimes \longrightarrow$	$\otimes$ = Addition				$\otimes$ = Multiplication			
		Hits				Hits		
Methods↓	MRR	@1	@3	@10	MRR	@1	@3	@10
MGIF w/o interaction feature	0.313	0.225	0.345	0.489	0.321	0.234	0.353	0.497
MGIF w/o data augmentation	0.358	0.266	0.393	0.544	0.357	0.266	0.39	0.542
MGIF	**0.361**	**0.269**	**0.398**	**0.546**	0.36	0.268	0.396	0.544

Table 8
Results of the Link Prediction by MRR and Hits@k on YAGO3-10 Dataset.

Models	YAGO3-10			
	MRR	Hits		
		@1	@3	@10
TransE [7]	0.238	0.212	0.361	0.447
DistMult [12]	0.340	0.240	0.380	0.540
ComplEx [15]	0.360	0.260	0.400	0.550
R-GCN [25]	0.235	0.212	0.376	0.421
ConvE [17]	0.440	0.350	0.490	0.620
RotatE [29]	0.487	0.398	0.545	0.667
TorusE [8]	0.481	0.387	0.521	0.640
HypER [32]	0.533	0.455	0.58	0.678
InteractE [18]	0.541	0.462	0.593	0.687
Rot-Pro [21]	0.542	0.443	0.596	**0.699**
MIF(ours)	**0.559**	0.484	**0.606**	0.694
MGRIF(ours)	0.557	**0.486**	0.600	0.689
MGEIF(ours)	0.555	0.479	0.602	0.693
MGIF(ours)	0.557	0.483	0.601	0.691

Table 9
Parameter amount analyses for the data augmentation module.

Model	Datasets				
	FB15k-237	WN18RR	WN18	FB15k	YAGO3-10
TransE	14.78M	20.47M	-	-	-
DistMult	29.56M	40.95M	-	-	-
ComplEx	29.56M	40.95M	-	-	-
RotatE	29.32M	40.95M	-	-	-
MGIF w/o data augmentation	11.2M	17.1M	17.1M	18.2M	35.7M
MGIF	13.7M	19.6M	19.6M	20.7M	38.2M
Parameter increase percentage	22.7%	14.9%	14.9%	13.9%	7.1%

4.2.4. Effect of data augmentation module

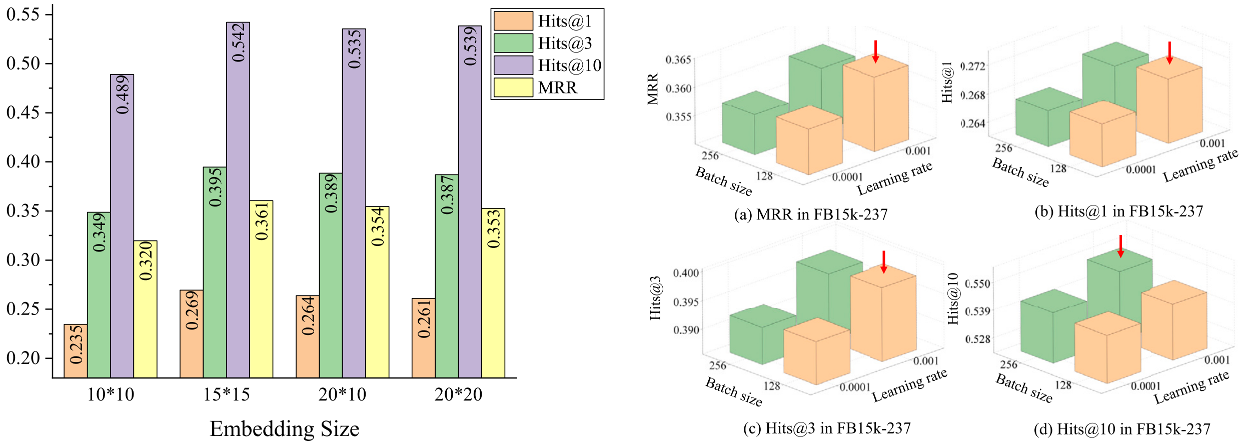
As shown in Table 9, although the data augmentation module of MGIF introduces several extra parameters to some extent. Even with the data augmentation module, the number of parameters in MGIF is still less than that of methods such as TransE, DistMult, ComplEx and RotatE. It is worth noting that the number of extra parameters introduced by the data augmentation module as a percentage of the total number of parameters in the model gradually decreases as the size of the KG increases. For example, on the small-scale KG WN18RR, the percentage of the number of parameters from the data augmentation module to the total number of parameters in the model is 14.9%. Still, on the larger-scale KG YAGO3-10, the percentage drops to 7.1%. This indicates that the additional number of parameters of the data augmentation module does not affect the application of the MGIF extension to the large-scale KG.

4.2.5. Effect of global feature composition

To explore how to leverage global features optimally and each MGIF module's efficiency, we tested two approaches to composing global features: addition and multiplication. We evaluated the impact of these two global feature composition types on different components of MGIF on FB15k-237. The results are shown in Table 7. MGIF without interaction features means applying only the features of global-aware convolution, and MGIF without data augmentation indicates no data augmentation component. The experimental results show that MGIF of the additive composite global features works best. MGIF is compared with MGIF without data augmentation to demonstrate that data augmentation can enhance model effectiveness by increasing interaction number. From

Table 10
MGIF running time (s) comparison for each epoch on five datasets.

Models	Space complexity $\mathcal{O}$	Datasets				
		FB15k	WN18	FB15k-237	WN18RR	YAGO3-10
ConvE	$(\vert\mathcal{E}\vert + \vert\mathcal{R}\vert)k$	33.10	34.89	17.68	22.86	217.03
HypER	$(\vert\mathcal{E}\vert + \vert\mathcal{R}\vert)k$	35.89	33.87	18.71	21.86	213.51
MGIF	$(\vert\mathcal{E}\vert + \vert\mathcal{R}\vert)k$	42.89	35.14	23.08	23.35	222.76



(a) MRR in FB15k-237

(b) Hits@1 in FB15k-237

(c) Hits@3 in FB15k-237

(d) Hits@10 in FB15k-237

(a) Embedding Dimension

(b) Batch Size and Learning Rate

Fig. 5. The Impact of Embedding Dimension, Batch Size, and Learning Rate on Experimental Results for the FB15k-237.

the experimental results of MGIF without interaction features, it is evidenced that incorporating interaction-aware features effectively improve the performance of MGIF.

4.2.6. Scalability analysis

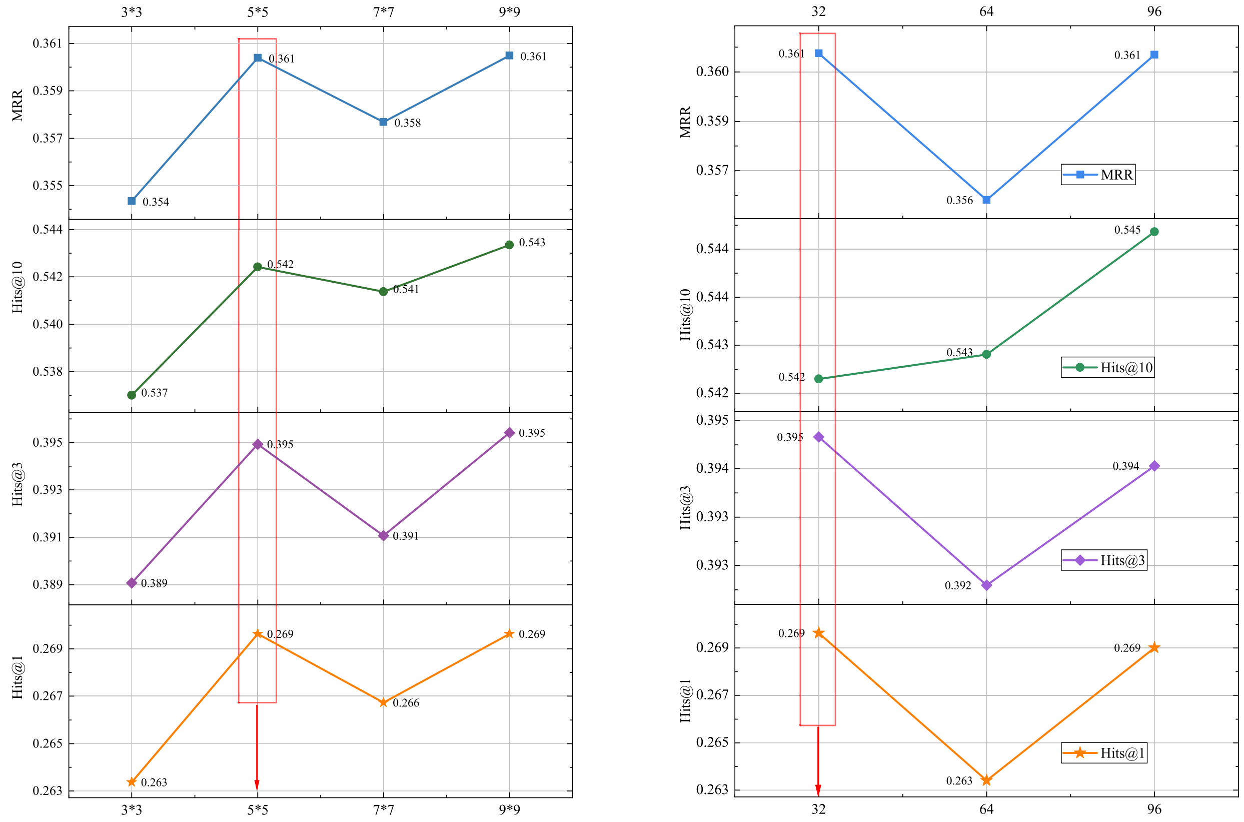
Real-world knowledge graphs tend to be enormous, and whether the model can efficiently and effectively model large-scale KGs reflects the model's scalability. YAGO3-10 contains more entities and relatively fewer relations than previous datasets, such as WN18 and FB15k-237. It comprises many triples, which requires the model to better distinguish between relations and entities. Meanwhile, this dataset has a more significant challenge than previous datasets. Experiments on the YAGO3-10 dataset yielded the experimental results in Table 8, where MIF is superior in all metrics compared to other methods. The explanation that the MGIF does not perform as well as MIF considering global information incorporations may be due to the excessive number of entities (relations) in YAGO3-10, thus resulting in too much clutter in the incorporated global information. It ultimately affects the MGIF's performance. Specifically, compared with ConvE, HypER, a method also CNN-based, MRR results are improved by 20%, Hits@1 boosted 3.1%, Hits@3 by 26%, 4%, and Hits@10 increased 17.5%, 2.4%, respectively. Compared to the optimal performance of other models, the MRR results are enhanced by 2.5%, Hits@1 improved by 5.3%, Hits@3, and Hit@10 by a slight enhancement. This proves that MIF can be more complex and model larger KGs.

4.2.7. Evaluation of model efficiency

Model efficiency is compared by the running time (s) in FB15k, FB15k-237, WN18, WN18RR, and YAGO3-10. The comparison models are of the same category, i.e., ConvE and HypER, and the GPU of the server used in the experiment is Tesla V100. Table 10 displays the experimental findings. The space complexity of all three models is almost $\mathcal{O}((|\mathcal{E}| + |\mathcal{R}|)k)$, and the time gap values are very close rather than proportionally enlarged on large and small datasets. It is not difficult to infer that the difference between MGIF and the same type of model will also be very small when extended to larger data sets. The results of this experiment demonstrate that the MGIF model has significant performance improvements at acceptable time complexity.

4.2.8. Parameter sensitivity analysis

Embedding dimension Dimension can affect the model's effectiveness. If the embedding dimension is low, the model will not fit well, while if the embedding dimension is high, the model will overfit. The hyperparameter adjustment sets four dimensions, i.e., 100, 200, 225, and 400 dimensions. Fig. 5(a) illustrates how the embedding dimension affected the experiment outcomes. It may be caused by the inadequate fitting of 10 × 10 due to the inclusion of too few features. The 20 × 20 metric also does not rank well because there are excessive features in a 400-dimensional vector, and the model suffers from overfitting. The difference between 15 × 15 and 20 × 10 embedding dimensions is not obvious, whereas 15 × 15 recomposes the embedding to an aspect ratio of 1, which makes the interaction between the convolutional kernel and the input embedding complete and balanced, resulting in better performance.



(a) Filter Size

(b) Filter Number

Fig. 6. Exploration of filter size and number for the FB15k-237 dataset.

Learning rate and batch size influence During the hyperparameter search for the batch size and learning rate, the remaining hyperparameters are held constant with model-specific configurations. In contrast, only the batch size and learning rate parameters are varied for the experiments. The evaluation metrics Hits@1, Hits@3, and MRR, as depicted in Fig. 5(b), demonstrate that the model achieves optimal performance with a batch size of 128 and a learning rate 0.001. Thus, the best parameters for the batch size and learning rate combination are 128 and 0.001.

Number of filter size The number of filter sizes also requires adjustment depending on the model and the dataset. Different convolution kernel sizes represent different numbers of relations head entities interactions during a single convolution operation. The preliminary results are displayed in Fig. 6(a), where 3×3 is the worst in all metrics, probably due to the limited interaction that can be performed with the small convolutional kernel, which results in the inability of the model to capture some of the more complex information. In addition, 7×7 is the second worst compared with 5×5, indicating that larger convolutional kernels may lead to overfitting. Finally, 9×9 is the best metric. However, it is not very different from 5×5, probably because the larger size of the convolutional kernel allows the model to interact over a large range, and such interaction overcomes the increase caused by overfitting. Finally, considering the time and space complexity, the 5×5 convolutional kernels were adopted.

Number of filter number settings The number of convolutional kernels also tends to impact the model's performance negatively. Since a few filters will fail to capture enough features, an excessive number of convolutional kernels will lead to the overfitting of the model. The commonly used parameters for the number of convolutional kernels are 32, 64, and 96. The relevant parameters are set for the experiments, and the experimental outcomes are displayed in Fig. 6(b). The convolutional kernels are set to 64 for each index, while MRR, hits@1, and hits@3 are better than 96 for 32, but hits@10 are higher than 32 for 96. Considering the time complexity and space complexity, the number of convolution kernels is set to 32.

5. Extended analysis and implications

5.1. Results analysis

MGIF and its vanilla achieve more satisfactory performance in most metrics than the comparison approach. In particular, MGIF has a faster and smoother convergence process compared to neural network models of the corresponding type. Complex relationship modelling and extensibility experiments demonstrate the generalizability and stability of MGIF and its vanilla. In conclusion, through many hyperparameter combination experiments, MGIF can effectively obtain more expressive knowledge graph embeddings while maintaining a satisfactory time and space consumption. The main advantages of our work have been revealed, mainly in the following aspects: (1) By modelling the global and interactive characteristics of entities, MGIF can effectively capture the multi-perspective knowledge that exists in KG, thus exhibiting more satisfactory performance. (2) Data enhancement methods and adaptive relation-aware convolution significantly boost the quantity and quality of entity-relation interactions. (3) MGIF and its variants have their advantages on different benchmarks, which provides an intuitive explanation for the stronger generalization ability of our proposed model.

6. Conclusions and future directions

Most of the entities (relations) have an unbalanced distribution in the process of finding losing entities in KGs. Multi-relation entities do not capture sufficiently generalized embedding representation from the training phase. Therefore, we design a multi-perspective knowledge graph completion model with global and interaction features (MGIF). The reshaped entities and relations are designed to interact at three views: relation-to-entity, entity-to-entity, and relation-to-relation, respectively. Therefore, multi-relation entities can perceive knowledge sharing that originates from multiple levels during the convolution of MGIF, which effectively alleviates the multi-domain semantic problem for multi-relation entities. During the model training, we provide a data augmentation approach that effectively increases the number of entity-relation interactions. In contrast, using adaptive relation-aware filters enables acquiring higher-quality interactions and expedites the model's convergence. Experimental results on five datasets with varying sparsity levels demonstrate the effective modelling of multi-relation entities by MGIF, achieving better performance for link prediction tasks.

However, we acknowledge that MGIF has its limitations. There is not only structural information in KGs but also multi-modal information. MGIF can only learn semantic structural information in KGs effectively. The ability to handle multi-modal information is what we need. Text and image information of entities and relations would be effective auxiliary information for learning KGE. In future research, we aim to integrate multi-modal information into MGIF to bridge the gap between diverse modalities of the same knowledge and augment the generalization capability of knowledge embedding.

CRediT authorship contribution statement

Duantengchuan Li: Conceptualization, Data curation, Methodology, Software, Supervision, Writing – original draft, Writing – review & editing. **Fobo Shi:** Conceptualization, Software, Writing – review & editing. **Xiaoguang Wang:** Data curation, Methodology, Supervision. **Chao Zheng:** Data curation, Resources, Writing – review & editing. **Yuefeng Cai:** Conceptualization, Methodology. **Bing Li:** Conceptualization, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62032016), Major Project of the National Social Science Fund, China (No. 21&ZD334), and the Key Research and Development Program of Hubei Province (No. 2021BAA031).

References

[1] H. Cui, T. Peng, F. Xiao, J. Han, R. Han, L. Liu, Incorporating anticipation embedding into reinforcement learning framework for multi-hop knowledge graph question answering, Inf. Sci. 619 (2023) 745–761.

[2] J. Gomes, R.C. de Mello, V. Ströele, J.F. de Souza, A hereditary attentive template-based approach for complex knowledge base question answering systems, Expert Syst. Appl. 205 (2022) 117725.

[3] H. Li, J. Wang, X. Du, Z. Hu, S. Yang, Kbhn: a knowledge-aware bi-hypergraph network based on visual-knowledge features fusion for teaching image annotation, Inf. Process. Manag. 60 (2023) 103106.

[4] J. Misztal-Radecka, B. Indurkhya, Bias-aware hierarchical clustering for detecting the discriminated groups of users in recommendation systems, Inf. Process. Manag. 58 (2021) 102519.

[5] M. Ahmadian, M. Ahmadi, S. Ahmadian, A reliable deep representation learning to improve trust-aware recommendation systems, in: Expert Systems with Applications, vol. 197, 2022, p. 116697.

[6] J. Wang, Q. Zhang, F. Shi, D. Li, Y. Cai, J. Wang, B. Li, X. Wang, Z. Zhang, C. Zheng, Knowledge graph embedding model with attention-based high-low level features interaction convolutional network, Inf. Process. Manag. (2023) 103350.

[7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-Relational Data, Advances in Neural Information Processing Systems, vol. 26, Curran Associates, Inc., 2013.

[8] T. Ebisu, R. Ichise, Toruse: knowledge graph embedding on a Lie group, in: Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 1819–1826.

[9] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 2015, pp. 687–696.

[10] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proc. 28th AAAI Conf. Artif. Intell., 2014, pp. 1112–1119.

[11] Z. Sun, Z. Deng, J. Nie, J. Tang, Rotate: knowledge graph embedding by relational rotation in complex space, in: Proc. 7th Int. Conf. Learn. Representations, 2019.

[12] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proc. 3rd Int. Conf. Learn. Representations, 2015.

[13] M. Nickel, V. Tresp, H. Kriegel, A three-way model for collective learning on multi-relational data, in: Proc. 28th Int. Conf. Mach. Learn., 2011, pp. 809–816.

[14] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, in: Proc. 30th AAAI Conf. Artif. Intell., vol. 30, 2016.

[15] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proc. 33rd Int. Conf. Mach. Learn., 2016, pp. 2071–2080.

[16] I. Balazevic, C. Allen, T. Hospedales TuckER, Tensor factorization for knowledge graph completion, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 5185–5194.

[17] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[18] S. Vashishth, et al., Interacte: improving convolution-based knowledge graph embeddings by increasing feature interactions, in: Proc. 34th AAAI Conf. Artif. Intell., vol. 34, 2020, pp. 3009–3016.

[19] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: representation, acquisition, and applications, IEEE Trans. Neural Netw. Learn. Syst. 33 (2022) 494–514.

[20] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proc. 29th AAAI Conf. Artif. Intell., 2015, pp. 2181–2187.

[21] T. Song, J. Luo, L. Huang Rot-pro, Modeling transitivity by projection in knowledge graph embedding, in: Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 24695–24706.

[22] X. Changyi, H. Xiangnan, C. Yixin, Knowledge graph embedding by normalizing flows, Proc. AAAI Conf. Artif. Intell. 37 (4) (2023) 4756–4764.

[23] L. Jin, Z. Yao, M. Chen, H. Chen, W. Zhang, A comprehensive study on knowledge graph embedding over relational patterns based on rule learning, in: The Semantic Web – ISWC 2023, 2023, pp. 290–308.

[24] K. Hayashi, K. Kishimoto, M. Shimbo, Binarized embeddings for fast, space-efficient knowledge graph completion, IEEE Trans. Knowl. Data Eng. 35 (2023) 141–153, https://doi.org/10.1109/TKDE.2021.3075070.

[25] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: Proc. 15th Extended Semantic Web Conf., 2018, pp. 593–607.

[26] S. Vashishth, S. Sanyal, V. Nitin, P. Talukdar, Composition-based multi-relational graph convolutional networks, in: Int. Conf. Learn. Represent., 2020.

[27] A. Zeb, S. Saif, J. Chen, A.U. Haq, Z. Gong, D. Zhang, Complex graph convolutional network for link prediction in knowledge graphs, Expert Syst. Appl. 200 (2022) 116796.

[28] Z. Zhang, J. Wang, J. Ye, F. Wu, Rethinking graph convolutional networks in knowledge graph completion, in: Proceedings of the ACM Web Conference 2022, WWW'22, 2022, pp. 798–807.

[29] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, B. Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, in: Proc. 33rd AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 3060–3067.

[30] Z. Li, Q. Zhang, F. Zhu, D. Li, C. Zheng, Y. Zhang, Knowledge graph representation learning with simplifying hierarchical feature propagation, Inf. Process. Manag. 60 (2023) 103348.

[31] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference, 2021, pp. 1737–1748.

[32] I. Balažević, C. Allen, T.M. Hospedales, Hypernetwork knowledge graph embeddings, in: Artif. Neural Networks Mach. Learn. – ICANN 2019: Workshop and Special Sessions, Springer, 2019, pp. 553–565.

[33] X. Jiang, Q. Wang, B. Wang, Adaptive convolution for multi-relational learning, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), 2019, pp. 978–987.

[34] Y. Fang, W. Lu, X. Liu, W. Pedrycz, Q. Lang, J. Yang, Circulare: a complex space circular correlation relational model for link prediction in knowledge graph embedding, IEEE/ACM Trans. Audio Speech Lang. Process. 31 (2023) 3162–3175.

[35] D. Li, T. Xia, J. Wang, F. Shi, Q. Zhang, B. Li, Y. Xiong, Sdformer: a shallow-to-deep feature interaction for knowledge graph embedding, Knowl.-Based Syst. 284 (2024) 111253.

[36] D. Jiang, R. Wang, L. Xue, J. Yang, Multisource hierarchical neural network for knowledge graph embedding, Expert Syst. Appl. 237 (2024) 121446.

[37] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proc. 5th Int. Conf. Artif. Intell. Stat, vol. 9, 2010, pp. 249–256.

[38] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, vol. 9, 2010, pp. 249–256.

[39] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd Int. Conf. Mach. Learn., vol. 37, 2015, pp. 448–456.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conf. Comput. Vis. Pattern Recog., 2016, pp. 2818–2826.

[42] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, 2015, pp. 57–66.

[43] L. Chao, J. He, T. Wang, W. Chu, PairRE: knowledge graph embeddings via paired relation vectors, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4360–4369.

[44] P. Baghershahi, R. Hosseini, H. Moradi, Self-attention presents low-dimensional knowledge graph embeddings for link prediction, Knowl.-Based Syst. 260 (2023).

[45] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.