



Research article

A lightweight CNN-based knowledge graph embedding model with channel attention for link prediction

Xin Zhou^{1,*}, Jingnan Guo¹, Liling Jiang¹, Bo Ning¹ and Yanhao Wang²

¹ School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

² School of Data Science and Engineering, East China Normal University, Shanghai 200062, China

* **Correspondence:** Email: zhouxin314159@dlnu.edu.cn; Tel: +8615840899025.

Abstract: Knowledge graph (KG) embedding is to embed the entities and relations of a KG into a low-dimensional continuous vector space while preserving the intrinsic semantic associations between entities and relations. One of the most important applications of knowledge graph embedding (KGE) is link prediction (LP), which aims to predict the missing fact triples in the KG. A promising approach to improving the performance of KGE for the task of LP is to increase the feature interactions between entities and relations so as to express richer semantics between them. Convolutional neural networks (CNNs) have thus become one of the most popular KGE models due to their strong expression and generalization abilities. To further enhance favorable features from increased feature interactions, we propose a lightweight CNN-based KGE model called IntSE in this paper. Specifically, IntSE not only increases the feature interactions between the components of entity and relationship embeddings with more efficient CNN components but also incorporates the channel attention mechanism that can adaptively recalibrate channel-wise feature responses by modeling the interdependencies between channels to enhance the useful features while suppressing the useless ones for improving its performance for LP. The experimental results on public datasets confirm that IntSE is superior to state-of-the-art CNN-based KGE models for link prediction in KGs.

Keywords: knowledge graph embedding; convolutional neural network; channel attention; link prediction; feature enhancement

1. Introduction

In recent years, large-scale knowledge bases such as FreeBase [1], DBpedia [2], YAGO [3], and WordNet [4] have been widely used in various domains, including information retrieval [5], question answering [6], and recommender systems [7]. Their powerful abilities for semantic processing and knowledge generalization have paved a new way for data management and mining. A knowledge

graph (KG) is a graph representation of a knowledge base in the form of a collection of fact triples (s, r, o) , each of which denotes a relation edge r between a head entity node s and a tail entity node o . The task of knowledge graph completion (KGC) is performed to improve the integrity of the KG which suffers from incompleteness by predicting the missing triples according to the facts. KGC can be performed by either extracting new facts from external sources, such as online encyclopedias and news wires [8], or by inferring them from those already in the KG. The latter approach, typically called Link Prediction (LP), is the focus of our paper.

Knowledge graph embedding (KGE) models have been shown to achieve the best performance for the task of link prediction in KGs among all the existing methods [9]. To learn low-dimensional vector or matrix representations of entities and relations in KGs, a lot of knowledge graph embedding models are proposed. Specifically, the classic triple-based embedding models are mainly divided into translation-based models (e.g., TransE [10], TransH [11], TransAH [12], TransR [13], TransD [14]), bilinear and tensor models (e.g., DistMult [15], SimpleE [16], TuckER [17]), neural network models (e.g., NTN [18], ER-MLP [19], ConvE [20], ConvKB [21], Conv-TransE [22], InteractE [23], DMACM [24], KMAE [25], JointE [26], CTKGC [27]), and complex vector models (e.g., ComplEx [28], RotatE [29], QuatE [30]).

Among all of the above models, convolutional neural network (CNN) based KGE models are attracting increasing research interest because they benefit from the explosion of deep learning techniques and exhibit strong expression and generalization abilities. The performance of CNN-based KGE models depends on the interactions between entities and relations. Considerable research efforts have been devoted to increasing the interactions between entities and relations. InteractE [23] increases interactions by embedding permutations. The directional multi-dimensional attention mechanism is designed to explore the deep expressive characteristic of the triple in DMACM [24]. Gaussian kernel and multi-attention techniques are adopted to expand the entities and relation embedding in KMAE [25]. JointE [26] utilizes 1D and 2D convolution operations jointly to extract the surface and explicit knowledge and increase the interactions between entities and relations. However, in order to improve performance, these models are complex with a large number of parameters. How to balance the efficiency and effectiveness of the CNN-based KGE model is still an open problem.

In this paper, we propose a lightweight CNN-based KGE model named IntSE with channel attention to improve the prediction accuracy for LP in KGs. Inspired by InteractE, checkered feature reshaping and circular convolution are adopted to increase the interactions between the components of entity and relation embeddings in IntSE. A channel attention module of IntSE is designed to enhance useful interactions while suppressing useless ones. IntSE is light and flexible, greatly reducing the number of parameters, but still has a competitive prediction accuracy for LP. The main contributions of this paper are summarized as follows:

- 1) We propose a lightweight CNN-based KGE model named IntSE to improve the prediction accuracy for LP in KGs. IntSE is efficient and effective for LP through three key components: checkered feature reshaping, circular convolution, and channel attention mechanism. Checkered feature reshaping and circular convolution operations are very effective for increasing interactions between entities and relations. Moreover, the channel attention mechanism further enhances the beneficial interactions.
- 2) In order to further improve the performance of link prediction, we propose a channel attention block called LPSENet, which improves the gate mechanism of the SENet prototype. LPSENet

improves the expression ability while reducing the model overfitting of IntSE.

- 3) We conduct extensive experiments on publicly available datasets to evaluate the performance of IntSE. The results show that IntSE achieves significantly higher accuracy for LP in KGs than the mainstream KGE models.

Paper organization: The rest of this paper is organized as follows. The existing literature related to this work is discussed in Section 2. Our proposed IntSE models for LP are presented in Section 3. The experimental setup and results are described in Section 4. The ablation studies which investigate how the checkered feature reshaping, circular convolution, and channel attention mechanism affect the performance of IntSE are discussed in Section 5. Finally, the whole paper is concluded in Section 6.

2. Related work

KGE learns an embedding representation of entities and relations of a KG in a continuous low-dimensional vector space. Wang et al. [31], Nguyen [9], and Liu et al. [32] systematically reviewed KGE models for knowledge graph completion (KGC). Moreover, Rossi et al. [33] and Akrami et al. [34] conducted experimental studies to assess the effectiveness of different link prediction (LP) methods on real-world data. According to the different ways of expressing the semantics between entities and relations, there are four families of KGE models: (a) TransE [10] and its families (such as TransH [11], TransAH [12], TransR [13], and TransD [14]) used the translation invariance to express the association between entities and relations; (b) DistMult [15] and its improved versions [16, 17] utilized a diagonal matrix to express the relations for model simplification; (c) ComplEx [28] and its variants [29, 30] further extended DistMult to the complex field, which could better model the asymmetric relations in KGs; and (d) Neural network (NN) based models (such as ER-MLP [19], NTN [13], ConvE [20], Conv-TransE [22], ConvKB [21], InteractE [23], DMACM [24], KMAE [25], JointE [26]) used neural network to model semantics between entity embeddings and relation embeddings. Among all the above methods, NN-based models demonstrated outstanding performance in prediction accuracy due to their excellent feature extraction ability. NN-based models for KGE, especially CNN-based models, have received more attention in recent years due to the advantages of convolution operation, such as parameter sharing, generalization, overfitting reduction, and robustness.

ConvE [20] is the first work on the CNN-based KGE model. It reshaped the head entity embedding and the relation embedding separately and stacked them into a 2D convolution layer to extract semantic features of entities and relations. The generated feature mapping was vectorized and projected into the embedding space of tail entities. Then, all candidate tail entity embedding vectors were matched through their inner products. Conv-TransE [22] considered that the reshaping of entity and relation embeddings in ConvE destroyed the translation invariance of the embedding vector. Thus, it removed the embedding vector reshape operation in ConvE and directly put the known entity and relation embeddings into the convolutional layer. The remaining procedure of Conv-TransE was the same as that of ConvE. Thus, Conv-TransE could be regarded as a variant of ConvE. ConvKB [21] was another CNN-based KGE model, which stacked the head entity embedding, relation embedding, and tail entity embedding into a matrix M , and used a convolution operation with filters $\omega \in \mathbb{R}^{1 \times 3}$ to act on each dimensional vector after M transposition. ConvKB was controversial since it obtained a competitive result on the FB15k-237 [35] dataset, but a disappointing result on the WN18RR [20]. As its performance varies with datasets and implementations, we do not compare it in our experiments.

InteractE [23] found that the stacked feature reshaping of ConvE (see Figure 1) limited the feature interaction between entities and relations. Thus, it increased the feature interaction between entity and relation embeddings to improve the performance for LP. Specifically, it reshaped the entity and relation embeddings into a checkered structure (see Figure 1), while using circular convolution to enhance the edge feature interaction. However, feature permutation in InteractE is costly. DMACM [24] found that the existing CNN-based KGE model ignored the directional relation characteristic and implicit fine-grained feature in the triple. Thus, it was proposed to explore the directional information and an inherent deep expressive characteristic of the triple by the directional multi-dimensional attention. Then, the output of directional multi-dimensional attention could be put in the convolution neural network with filters $\omega \in \mathbb{R}^{1 \times 3}$. KMAE [25] first expanded the entity embedding and relation embedding into entity kernel and relation kernel using the Gaussian kernel function. Entity kernel or relation kernel combined relation embedding or entity embedding input into a 2D convolutional layer, then two groups of channel attention and spatial attention captured high-quality feature vector information. JointE [26] consisted of path 1 and path 2 in two feed-forward paths. Path 1 used 1D convolution filters over input the entity and relation embedding to extract the surface and explicit knowledge. To reduce the number of parameters, path 2 employed different 2D convolution filters to extract the deep features from the reshaped entity and relation embedding. The final features of the entity and relation were obtained by element-wise addition between the output of path 1 and the output of path 2.

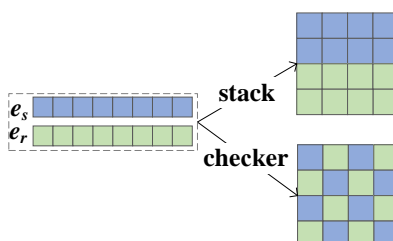


Figure 1. Different types of reshaping methods in KGE models [23].

In conclusion, CNN-based KGE models should expand feature extraction capability and increase the interactions between the entities and relations to improve the performance of LP. The key components of existing methods include the initial embedding of the entities and relation, the reshaping plan of the entities and relation embeddings, the convolution operations to extract features, and the attention-based feature calibration. Although many complex models can improve the performance for LP, they introduce too many parameters that hinder the model's efficiency. Thus, our goal is to design a CNN-based KGE model that can strike a better balance between performance and cost.

3. Our model

In this section, we introduce the proposed model IntSE. We first introduce the related notations and problem definition and then describe the architecture of IntSE in detail, as shown in Figure 2.

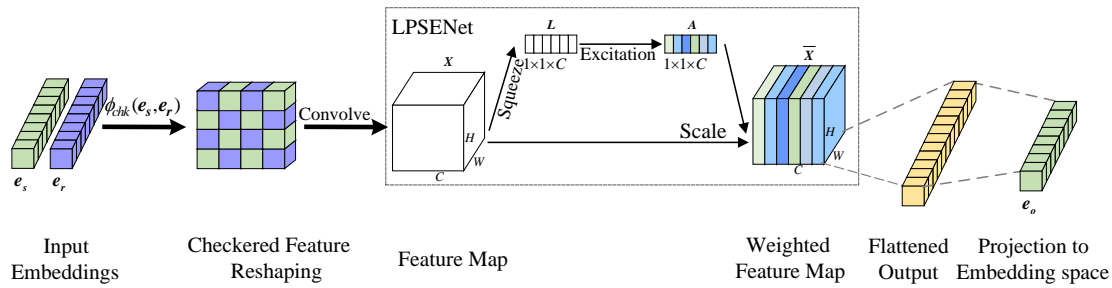


Figure 2. The architecture of IntSE.

3.1. Notations and problem definition

Let \mathcal{E} and \mathcal{R} denote the set of entities and relation types. A Knowledge Graph (KG) \mathcal{G} consists of a set of fact triples (s, r, o) , which is formally expressed as follows:

$$\mathcal{G} = \{(s, r, o) \mid s, o \in \mathcal{E}, r \in \mathcal{R}\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$$

Link Prediction (LP) is to predict the missing entity in a triple (s, r, o) , such as predicting o for a given $(s, r, ?)$ or s for a given $(?, r, o)$. LP can be formalized as a single sample learning-to-rank problem. A KGE-based LP method has two key components, namely encoding and scoring. The encoding component maps the head entity s , relation r , and tail entity o to the d -dimensional vector representations $e_s, e_r, e_o \in \mathbb{R}^d$. Then, the scoring component measures the authenticity of triples. The goal of LP is to learn a scoring function ψ about entity and relation embeddings so that the score $\psi(s, r, o)$ of fact triples (s, r, o) is higher than that $\psi(s', r, o')$ of non-fact triples (s', r, o') .

3.2. IntSE: A lightweight CNN-based KGE model with channel attention

The architecture of our IntSE model is shown in Figure 2. The key components of IntSE are checkered feature reshaping, circular convolution operation, and channel attention mechanism. Given a d -dimensional entity embedding vector e_s and a d -dimensional relation embedding vector e_r , IntSE first reshapes e_s and e_r into a checkered matrix $\phi_{chk}(e_s, e_r) \in \mathbb{R}^{m \times n}$, where $m \times n = 2d$, and then sends it to a 2D circular convolutional layer. Then, the convolutional layer outputs a feature map tensor $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_C] \in \mathbb{R}^{H \times W \times C}$, which is sent to the channel attention block for feature calibration to generate the calibrated feature map tensor $\bar{\mathbf{X}} \in \mathbb{R}^{H \times W \times C}$. Next, $\bar{\mathbf{X}}$ is vectorized into $vec(\bar{\mathbf{X}}) \in \mathbb{R}^{HWC}$, projected to the d -dimensional vector space using the linear transformation of the parameter matrix as $\mathbf{W} \in \mathbb{R}^{HWC \times d}$, and finally matched with the tail entity e_o via the inner product operation.

3.2.1. Checkered feature reshaping and circular convolution operation

The checkered feature reshaping increases the feature interaction between the components of entity and relation embeddings, and thus improves the expression ability of CNNs. Different from the stacked feature reshaping (see Figure 1), a checkered structure (see Figure 1) arranges the entity and relation embeddings such that no two adjacent cells are occupied by components of the same embedding to increase the feature interaction between entity and relation embeddings. The components of entity

and relation embedding are randomly assigned to cells of a checkered structure in IntSE. Then, a 2D circular convolutional layer with filters $\omega \in \mathbb{R}^{3 \times 3}$ is adopted to extract the deep and latent knowledge from the checkered features. Circular convolution induces more interactions than standard convolution and thus further expands the expression ability of IntSE.

In this paper, we are inspired by InteractE and borrow the checkered feature reshaping and circular convolution components to IntSE. Unlike the feature enumerations of InteractE, which take up a large number of parameters, IntSE only uses one checkered feature reshaping and one circular convolutional layer to improve the efficiency of the model.

3.2.2. Channel attention

SENet [36] explicitly defines the association between feature channels and filters the feature channel information by self-learning the weight of each channel, so as to enhance useful features and suppress useless features. We are inspired by SENet and related works, SENet is added to IntSE to enhance useful interactions and improve the performance of LP. However, the improvement is still quite limited. Therefore, we further propose a new SENet variant called LPSENet as shown in Figure 3 to enhance the useful features for LP.

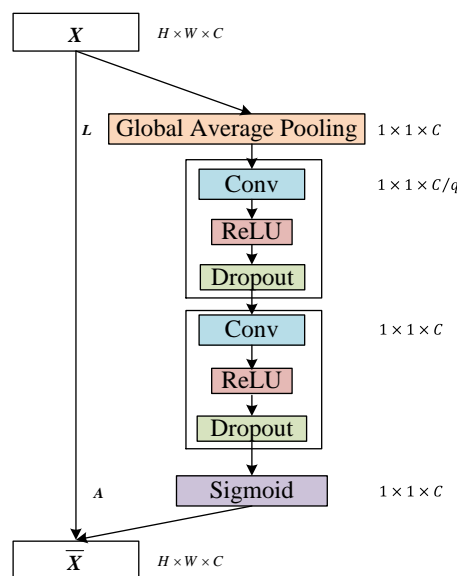


Figure 3. The diagram of improved SENet module: LPSENet.

For a feature map tensor $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_C] \in \mathbb{R}^{H \times W \times C}$, where \mathbf{x}_i represents the feature map of i -th channel of size $H \times W$, and C denotes the total number of channels. The operation flow of LPSENet is as follows:

- 1) Squeeze: LPSENet performs the Global Average Pooling (GAP) on each feature channel \mathbf{x}_c to obtain statistical information as l_c in the following:

$$l_c = p(\mathbf{X}) = \frac{1}{WH} \sum_{i=1, j=1}^{W, H} \mathbf{X}_{ij}. \quad (3.1)$$

The total C statistical values are aggregated into a compressed feature tensor $\mathbf{L} = [l_1, l_2, \dots, l_C] \in \mathbb{R}^{1 \times 1 \times C}$, which contains the global information of all features.

- 2) Excitation: To capture the dependence between channels, the gating mechanism is used to generate the weight matrix $\mathbf{A} = [a_1, a_2, \dots, a_C] \in \mathbb{R}^{1 \times 1 \times C}$ corresponding to each feature channel. Unlike the gating mechanism of SENet, we design a lighter and more efficient gating mechanism. SENet uses a gating mechanism consisting of a fully connected dimension-reduction layer, a ReLU [37] activation layer, and a fully connected dimension-increasing layer. The fully connected layer has a huge number of parameters and is prone to overfitting. Thus, in the LPSENet, the fully connected layer is changed to the convolutional layer with filters $\omega \in \mathbb{R}^{1 \times 1}$, and Dropout is added to avoid overfitting. Equations (3.2) and (3.3) formalize the gate mechanism of SENet and LPSENet, respectively.

$$\mathbf{A} = \sigma(\mathbf{W}_2 f(\mathbf{W}_1 \mathbf{L})), \quad (3.2)$$

$$\mathbf{A} = \sigma(f(f(\mathbf{L} \star \omega_1) \star \omega_2)), \quad (3.3)$$

where f is the ReLU function and σ is the Sigmoid function, $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{q} \times C}$ and $\mathbf{W}_2 \in \mathbb{R}^{C \times \frac{C}{q}}$ are learned to explicitly model the correlation between feature channels, and q is the reduction ratio, ω_1 and ω_2 are the convolution filters.

Compared with the SENet, LPSENet introduces more non-linearity. Thus, it has a higher expressive ability and is more robust to overfitting. The convolutional layer has the same output effect as the fully connected layer, but reduces the number of parameters and improves the computational efficiency. The experimental results in Section 5 will show that the LPSENet is more suitable for LP than SENet.

- 3) Scale: The output weight \mathbf{A} of the excitation operation represents the importance of each feature channel after feature selection, and the final output of the LPSENet block $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_C] \in \mathbb{R}^{H \times W \times C}$ is obtained by rescaling the input feature map \mathbf{X} with the activations:

$$\bar{\mathbf{x}}_c = a_c \cdot \mathbf{x}_c \quad (3.4)$$

where $\bar{\mathbf{x}}_c$ refers to channel-wise multiplication between the scalar a_c and the feature map \mathbf{x}_c .

3.2.3. Scoring function

The goal of LP is to learn a scoring function ψ about entity and relation embeddings so that the score $\psi(s, r, o)$ of fact triples (s, r, o) is higher than that $\psi(s', r, o')$ of non-fact triples (s', r, o') . Table 1 summarizes the scoring functions of CNN-based KGE models, where \bar{e}_s and \bar{e}_r denote the 2D stack reshaping of the entity embedding e_s and relation embedding e_r , $[\cdot]$ represents the concatenation operation, \star denotes the standard convolution operation, ω denotes the set of convolutional filters, \mathbf{W} and \mathbf{w} are the learned weight matrix, $\phi_{chk}(\mathcal{P}_k(e_s, e_r))$ denotes k different checkered reshaping of e_s and e_r , \clubsuit denotes the circular convolution, $vec(\cdot)$ represents the vectorization function, f and g are non-linear functions, $DM(\cdot)$ represents the directional multi-dimensional attention functions of DMACM [24], $kernel(\cdot)$ and $Mul(\cdot)$ represent the Gaussian kernel functions and multi-attention in KMAE [25], $ca(\cdot)$ represents the channel attention functions of IntSE.

Table 1. Scoring functions of CNN-based KGE models.

| Model | fScoring Function $\psi(e_s, e_r, e_o)$ |
|------------------|---|
| ConvE [20] | $f(\text{vec}(f([\bar{e}_s; \bar{e}_r] \star \omega))\mathbf{W})e_o$ |
| Conv-TransE [22] | $g(\text{vec}(f([e_s; e_r] \star \omega))\mathbf{W})e_o$ |
| ConvKB [21] | $\text{concat}(g([e_s; e_r; e_o] \star \omega)\mathbf{w})$ |
| InteractE [23] | $g(\text{vec}(f(\phi_{chk}(\mathcal{P}_k(e_s, e_r)) \star \omega))\mathbf{W})e_o$ |
| DMACM [24] | $\text{concat}(f(\text{DM}(e_s, e_r, e_o)) \star \omega)\mathbf{W}$ |
| KMAE [25] | $g(\text{Mul}([\text{kernel}(\bar{e}_s); \bar{e}_r] \parallel [\bar{e}_s; \text{kernel}(\bar{e}_r)]))e_o$ |
| JointE [26] | $f([e_s; e_r] \star \omega_{1D}^1 \star \omega_{1D}^2 + \text{vec}([\bar{e}_s \star \omega_r; \bar{e}_r \star \omega_s])\mathbf{W})e_o$ |
| IntSE | $g(\text{vec}(ca(f(\phi_{chk}(e_s, e_r) \star \omega)))\mathbf{W})e_o$ |

3.2.4. Training procedure of IntSE

In order to train the model parameters, we use the binary cross entropy with label smoothing as the loss function [20], as shown in Eq (3.5):

$$\Gamma(p, t) = -\frac{1}{N} \sum_i (t_i \cdot \log_2(p_i) + (1 - t_i) \cdot \log_2(1 - p_i)), \quad (3.5)$$

where $p = \sigma(\psi(s, r, o))$ indicates the probability of correct link prediction through applying the logistic sigmoid function $\sigma(\cdot)$ to the scores $\psi(s, r, o)$ and t is the smoothing label. In this paper, we use Adam [38] as the optimizer and label smoothing to reduce the overfitting due to the saturation of output non-linearity at the labels [39].

4. Experiments

In this section, we first describe the setup of our experiments. Then, we present the detailed experimental results and provide thorough analyses.

4.1. Experimental setup

Dataset: We use the two most widely used knowledge graph datasets for link prediction, namely FB15k-237 [35] and WN18RR [20], in our experiments. Specifically, FB15k-237 is an improved version of the FB15k [10] dataset derived from FreeBase [1], where all reversal relations in FB15k are deleted to prevent test triples from being directly deduced by reverse training triples. WN18RR is a subset of the WN18 [10] dataset derived from WordNet [4], and the inverse relationships are also deleted similar to FB15k-237. The characteristics of the two datasets are shown in Table 2.

Table 2. Statistics of FB15k-237 and WN18RR dataset.

| Dataset | #entities | #relations | #train | #valid | #test |
|-----------|-----------|------------|---------|--------|--------|
| FB15k-237 | 14,541 | 237 | 272,115 | 17,535 | 20,466 |
| WN18RR | 40,943 | 11 | 86,835 | 3034 | 3134 |

Evaluation metrics: In this paper, we use the four commonly used evaluation criteria for link prediction, namely, MR, MRR, Hits@1, and Hits@10, to measure the performance of different models.

The mean rank (MR) represents the mean of the test triples' ranks, i.e.,

$$MR = \frac{1}{2|T|} \sum_{(s,r,o) \in T} (rank_s + rank_o),$$

where $|T|$ is the size of the test set. The mean reciprocal rank (MRR) is the average inverse of the harmonic mean of the test triples' ranks, i.e.,

$$MRR = \frac{1}{2|T|} \sum_{(s,r,o) \in T} \left(\frac{1}{rank_s} + \frac{1}{rank_o} \right).$$

And Hits@ k is the percentage of top- k results that are correct. Among the four indicators, lower MR and higher MRR, Hits@1, and Hits@10 indicate better performance.

Hyperparameter setting and implementation: The ranges of hyperparameters are set as follows: the learning rate $\gamma \in \{0.01, 0.001, 0.005, 0.0001\}$, the entity and relation embedding dimension $d \in \{100, 200\}$, the kernel size in the 2D convolutional layer $k \in \{5, 7, 9, 11\}$, the reduction ratio of LPSENet $1/q$, where $q \in \{2, 4, 8, 16\}$. In addition, the experiment also uses batch normalization and Dropout to avoid overfitting. To decide the values of hyperparameters, each model is trained for 500 epochs via the grid search method, and the hyperparameters with the best performance are selected according to the MRR on the validation set [20, 23]. For the FB15k-237 dataset, the parameter settings are $\gamma = 0.0001$, $d = 200$, $k = 9$, and $q = 4$, and the batch size is 128. For the WN18RR dataset, the parameter settings are $\gamma = 0.001$, $d = 200$, $k = 11$, and $q = 8$, and the batch size is 256. We implemented our models with PyTorch in Python 3. All experiments were conducted on a server with 4 Intel® Xeon® w-2104 CPUs @ 3.20GHz, 32GB DDR4 RAM, and 1 NVIDIA® GeForce® RTX™ 2060 GPU with 8GB GDDR6 RAM.

4.2. Experimental results and analyses

In the experiments, we compare IntSE with the six baseline models, namely ConvE [20], ConvTransE [22], InteractE [23], DMACM [24], KMAE [25], JointE [26].

4.2.1. Prediction accuracy

Table 3. Prediction accuracy on the FB15k-237 and WN18RR dataset.

| Model | FB15k-237 | | | | WN18RR | | | |
|------------------|--------------|------------|--------------|--------------|--------------|-------------|--------------|--------------|
| | MRR | MR | Hits@10 | Hits@1 | MRR | MR | Hits@10 | Hits@1 |
| ConvE [20] | 0.325 | 244 | 0.501 | 0.237 | 0.430 | <u>4187</u> | 0.520 | 0.400 |
| Conv-TransE [22] | 0.330 | - | 0.510 | 0.240 | 0.460 | - | 0.520 | 0.430 |
| InteractE [23] | 0.354 | 172 | 0.535 | <u>0.263</u> | 0.463 | 5202 | 0.528 | 0.430 |
| DMACM [24] | 0.270 | 244 | 0.440 | - | 0.230 | 552 | 0.540 | - |
| KMAE [25] | 0.326 | 235 | 0.502 | 0.240 | 0.448 | 4441 | 0.524 | 0.415 |
| JointE [26] | <u>0.356</u> | <u>177</u> | 0.543 | 0.262 | 0.471 | 4655 | <u>0.537</u> | <u>0.438</u> |
| IntSE | 0.359 | 179 | <u>0.540</u> | 0.267 | <u>0.469</u> | 5007 | 0.532 | 0.439 |

The prediction accuracy of different models on the FB15k-237 and WN18RR datasets is shown in Table 3. The scores of all the baselines are taken directly from the values reported in the original

papers. The best and second-best results are highlighted by bold and underlined texts, respectively. We observe that IntSE achieves high accuracy in both datasets: it achieves competitive performance in all evaluation metrics and outperforms all the baselines in terms of MRR and Hits@1 on FB15k-237. Specifically, we draw the following conclusions from the experimental results:

- 1) IntSE outperforms InteractE in terms of MRR, Hits@10, and Hits@1 on both datasets, which indicates that channel attention effectively improves the performance of the CNN-based KGE model for LP. Furthermore, IntSE obtains improvements of 2.5 and 3% over InteractE in terms of MRR and Hits@1 on the FB15k-237 dataset. Then, on the WN18RR dataset, the improvement rates of IntSE over InteractE on MRR and Hits@1 are 1.7 and 2.3%. These results confirm that IntSE not only retains the advantages of InteractE but also further enhances the useful features of InteractE.
- 2) Both IntSE and InteractE outperform KMAE in terms of MRR, Hits@10, and Hits@1 on both datasets, which indicates that checkered feature reshaping and circular convolution operations increase feature interactions between the entities and relations are effective. The ablation studies of IntSE in Section 5 further demonstrate the effectiveness of checkered feature reshaping and circular convolution operations.
- 3) It is unexpected that DMACM outperforms all the models in the MR and Hits@10 on the WN18RR dataset. However, DMACM is inferior to others models in the MRR metrics on both datasets. The reason may be the distinguishing feature of WN18RR, which has fewer relations but more entities.
- 4) Compared to JointE, the state-of-the-art CNN-based KGE model, IntSE performs better in terms of Hits@1 on both datasets. Moreover, they are well-matched in other metrics.

4.2.2. Prediction accuracy for different relation types

Table 4. Prediction accuracy by relation category on the FB15k-237 dataset.

| | | ConvE [20] | | InteractE [23] | | IntSE | |
|-----------|------------------------|------------|---------|----------------|---------|--------------|--------------|
| | | MRR | Hits@10 | MRR | Hits@10 | MRR | Hits@10 |
| Head Pred | 1-to-1 | 0.374 | 0.505 | 0.386 | 0.547 | 0.391 | 0.561 |
| | 1-to- <i>n</i> | 0.091 | 0.170 | 0.106 | 0.192 | 0.112 | 0.198 |
| | <i>n</i> -to-1 | 0.444 | 0.644 | 0.466 | 0.647 | 0.472 | 0.652 |
| | <i>n</i> -to- <i>m</i> | 0.261 | 0.459 | 0.276 | 0.476 | 0.285 | 0.479 |
| Tail Pred | 1-to-1 | 0.366 | 0.510 | 0.368 | 0.547 | 0.372 | 0.550 |
| | 1-to- <i>n</i> | 0.762 | 0.878 | 0.777 | 0.881 | 0.780 | 0.888 |
| | <i>n</i> -to-1 | 0.069 | 0.150 | 0.074 | 0.141 | 0.079 | 0.152 |
| | <i>n</i> -to- <i>m</i> | 0.375 | 0.603 | 0.395 | 0.617 | 0.400 | 0.621 |

To further verify the effectiveness of IntSE, we also evaluate the performance of the models for LP in different categories of relations on the FB15k-237 dataset. We use FB15k-237 in this set of experiments because its relation is more diverse. Based on the average number of tail entities per head entity and the average number of head entities per tail entity, the relation is divided into four categories: 1-to-1, 1-to-*n*, *n*-to-1, and *n*-to-*m*. Here, an average number of less than 1.5 is marked as “1” and “*n*” otherwise. Among the 224 distinct relations in the test set of FB15k-237, 5.8% are 1-to-1, 11.6% are

1-to- n , 33.9% are n -to-1, and 48.7% are n -to- m relations. But on the WN18RR dataset, the 11 distinct relations in the test set are distributed as 2, 4, 3, and 2 in these four classes [34]. Because all the other baselines don't report the performance of LP in different relation categories in their papers, ConvE [20] and InteractE [23], two classical CNN-based KGE models, are used for comparison in the experiments. MRR and Hits@10 are used as evaluation criteria [14]. Table 4 presents the experimental results of different models for LP on the FB15k-237 dataset. From Table 4, we find that IntSE achieves better performance than ConvE and InteractE in all four relation categories, whether it deals with simple relation categories (e.g., 1-to-1) or more complex relation categories (e.g., 1-to- n and n -to- m). It is verified again that IntSE has good robustness and is suitable for link prediction tasks with various relation categories.

4.2.3. Prediction accuracy by feature permutation

IntSE only uses one checkered feature reshaping and one circular convolutional layer; that is, the number of feature permutation $perm$ in IntSE is one. We conduct experiments in the same setting to further evaluate the impact of feature permutation on the prediction accuracy of InteractE and IntSE. Table 5 presents the experimental results. We can see that the prediction accuracy of InteractE and IntSE drops slowly with the increase of $perm$ on the FB15k-237 dataset. IntSE and InteractE work best when $perm$ is one. This result is consistent with that reported in the paper InteractE [23]. Nevertheless, IntSE is still superior to InteractE in all cases under all the metrics. From Table 5, we can further conclude that increasing the beneficial feature interactions is crucial to improve the performance of CNN-based KGE models.

Table 5. Prediction accuracy with varying feature permutation on the FB15k-237 dataset.

| | InteractE [23] | | | IntSE | | |
|-----------------|----------------|------------|------------|--------------|------------|------------|
| | $perm = 1$ | $perm = 2$ | $perm = 3$ | $perm = 1$ | $perm = 2$ | $perm = 3$ |
| MRR | 0.353 | 0.353 | 0.349 | 0.354 | 0.353 | 0.350 |
| Hits@10 | 0.537 | 0.534 | 0.533 | 0.539 | 0.537 | 0.535 |
| Hits@20 | 0.619 | 0.616 | 0.613 | 0.620 | 0.615 | 0.614 |
| Hits@40 | 0.694 | 0.692 | 0.688 | 0.695 | 0.691 | 0.688 |
| Hits@80 | 0.764 | 0.761 | 0.759 | 0.766 | 0.763 | 0.758 |
| Hits@160 | 0.829 | 0.824 | 0.822 | 0.830 | 0.827 | 0.822 |

4.2.4. Parameter efficiency

We also evaluate the parameter efficiency of different models on the FB15k-237 dataset. Table 6 presents the experimental results. The number of parameters in IntSE is larger than ConvE because IntSE adopts 96 convolution filters of size 9×9 while ConvE adopts 32 convolution filters of size 3×3 . The fully-connected layer of IntSE has much more parameters than ConvE. The number of parameters in IntSE is nearly equal to that of InteractE when the feature permutation $perm$ is one. The channel attention mechanism of IntSE is lightweight with only a few thousand additional parameters. The number of parameters in InteractE increases significantly with the feature permutation $perm$ because the number of parameters in the convolution and fully-connected layers of InteractE is positively correlated with $perm$.

Table 6. Parameter efficiency of different models on the FB15k-237 dataset.

| Model | ConvE [20] | InteractE [23] | | | | IntSE |
|------------|--------------|----------------|----------|----------|----------|-------|
| | | perm = 1 | perm = 2 | perm = 3 | perm = 4 | |
| Parameters | 4.96M | <u>10.7M</u> | 18.38M | 26.07M | 33.75M | 10.7M |

5. Ablation studies

There are three key components in IntSE: checkered feature reshaping, circular convolution operation, and channel attention mechanism. Based on the public FB15k-237 dataset, ablation studies are conducted to evaluate the effects of these key components on the performance of IntSE. Specially, we evaluate the effects of the three different channel attention mechanisms on the performance of IntSE.

5.1. Key components of IntSE

The effects of key components on the performance of IntSE are shown in Table 7. The *sr*, *cr*, *sc*, *cc*, and *ca* denote the stacked feature reshaping, checkered feature reshaping, standard convolution, circular convolution, and channel attention components, respectively. From Table 7, we can see that checkered feature reshaping, circular convolution operation, and channel attention are three components that significantly contribute to the performance of IntSE. The channel attention mechanism significantly boosts performance in all cases. Although *cr* + *cc* outperforms most variants of IntSE with the help of checkered feature reshaping and circular convolution, it is inferior to IntSE. IntSE obtains improvements of 4 and 3.5% over *cr* + *cc* in terms of MRR and Hits@1 with the help of the channel attention mechanism. These findings further demonstrate the effectiveness of IntSE and the importance of the channel attention mechanism.

Table 7. Effect of key components on the performance of IntSE.

| Model | MRR | MR | Hits@10 | Hits@1 |
|------------------|--------------|------------|--------------|--------------|
| sr+sc (ConvE) | 0.325 | 244 | 0.501 | 0.237 |
| sr+sc+ca | 0.342 | 181 | 0.525 | 0.251 |
| cr+sc | 0.338 | 185 | 0.519 | 0.249 |
| cr+sc+ca | 0.350 | 193 | 0.536 | 0.262 |
| cr+cc | 0.346 | 175 | 0.532 | 0.258 |
| cr+cc+ca (IntSE) | 0.359 | 179 | 0.540 | 0.267 |

5.2. Different channel attention mechanisms

The channel attention mechanism is the key component of IntSE. To further confirm the necessity of the channel attention mechanism, we evaluate the effects of the three different channel attention mechanisms on the performance of IntSE. Specifically, SENet [36] is the first work to boost the representation power of a CNN by enhancing channel relationship; SKNet [40] is the first to explicitly focus on the adaptive receptive field size of neurons by introducing the attention mechanism; ECA-Net [41] is the most effective channel attention module of deep CNNs in computer vision applications. Table 8 presents the detailed results of IntSE variants with different channel attention mechanisms under the four evaluation indicators: MR, MRR, Hits@1, and Hits@10.

Table 8. Performance of IntSE with different channel attention mechanisms.

| Model | MRR | MR | Hits@10 | Hits@1 |
|---------------|--------------|------------|--------------|--------------|
| IntSE-SENet | 0.358 | 183 | 0.538 | 0.266 |
| IntSE-SKNet | 0.350 | 193 | 0.527 | 0.262 |
| IntSE-ECANet | 0.346 | 223 | 0.529 | 0.250 |
| IntSE-ECANet' | 0.355 | 184 | 0.536 | 0.263 |

IntSE-SENet performs the best among all models compared. To our surprise, ECA-Net brings side effects on IntSE. Especially the IntSE-ECANet model, which integrates checkered feature reshaping, circular convolution operation, and ECA-Net, has the lowest performance. The main reason is that a KG has different data characteristics from images and video. The IntSE model is so simple that there is only one convolution layer with a small input size, while deep CNNs in computer vision applications often have very large input sizes. Detailed analyses of the experimental results lead to the following views in the CNN-based KGE models for LP:

- 1) *The channel attention is more important than the spatial attention.* SKnet uses soft attention to fuse the features of multiple convolution branches with different kernel sizes. The fact that IntSE outperforms IntSE-SKNet for LP indicates that the convolution kernel size of IntSE is more appropriate. Moreover, since IntSE-SENet outperforms IntSE-SKNet, we again confirm that channel attention is more important than spatial attention. The experimental results in [42] also show that the channel-first order is slightly better than the spatial-first order. Therefore, we should pay more attention to explicitly defining the association between feature channels so as to improve the performance for LP.
- 2) *The global cross-channel interaction is more important than local cross-channel interaction.* Although ECA-Net is an efficient and effective channel attention mechanism for deep CNNs, the performance of IntSE-ECANet is lower than IntSE-SENet. The possible reason is that a fast 1D convolution with kernel size k in ECA-Net only captures the local cross-channel interaction rather than the global cross-channel interaction. To verify the hypothesis, we conduct additional experiments on a model named IntSE-ECANet' which adopts a variant of ECANet with only a single fully connected layer. Since IntSE-ECANet' exhibits better performance than IntSE-ECANet, we further confirm that the global cross-channel interaction is more important than the local cross-channel interaction in CNN-based KGE models for LP. We observe that IntSE-SENet performs the best among all models compared. This indicates that the two fully connected layers designed in SENet to capture the non-linear global cross-channel interactions are effective, and we can improve the performance of the CNN-based KGE model on LP by introducing SENet.

5.3. Different gating mechanisms of SENet

The channel attention module of IntSE is LPSENet which changes the gating mechanism of SENet. In the LPSENet, the fully connected layer is changed to the convolutional layer with filters $\omega \in \mathbb{R}^{1 \times 1}$, and Dropout is added to avoid overfitting. We also evaluate the performance of IntSE with different gating mechanisms. The results are shown in Table 9. We can see that IntSE achieves better performance with the gating mechanism of LPSENet. The reason is that it has a higher expressive ability and more robustness to overfitting.

Table 9. Performance of IntSE with different gating mechanisms in channel attention.

| Model | MRR | MR | Hits@10 | Hits@1 |
|-------------|--------------|-------------|--------------|--------------|
| IntSE-SENet | 0.466 | 5123 | 0.530 | 0.434 |
| IntSE | 0.469 | 5007 | 0.532 | 0.439 |

6. Conclusion remarks

A lightweight CNN-based knowledge graph embedding (KGE) model with channel attention called IntSE is proposed in this paper. Although CNN-based KGE models attract more attention from the research and achieve higher LP accuracy than other KGE models, they often contain too many parameters and have very low efficiency. We do our utmost to explore the balance between the efficiency and effectiveness of the CNN-based KGE model and propose IntSE through extensive experiments. The key idea of IntSE is to increase the favorable feature interactions between the entities and relations in a triple of KG. Checkered feature reshaping and circular convolution operations increase the feature interactions, and the channel attention further enhances useful feature interactions in IntSE. The ablation studies are carried out to explore the effectiveness of each component on the IntSE for LP. Especially the impacts of different channel attention mechanisms (i.e., SENet, SKNet, and ECA-Net) on IntSE are investigated. Compared with the state-of-the-art CNN-based KGE models, IntSE mostly achieved the best performance under various evaluation criteria on public datasets. These extensive experimental results substantiate the efficiency and effectiveness of our model.

In future work, we would like to explore whether other improved versions of SENet can further improve the performance of InteractE and investigate whether incorporating the spatial attention module into IntSE can boost its performance. In addition, it would also be interesting to consider how to improve the performance of KGE models for other downstream tasks, e.g., question answering [6], recommendation [7], and entity resolution [31].

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant (No. 61976032 and No. 62002039), the Fundamental Research Funds for the Central Universities (No. 3132022261 and No. 3132022634).

Conflict of 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.

References

1. K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, (2008), 1247–1250. <https://doi.org/10.1145/1376616.1376746>

2. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, et al., DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, *Semant. Web*, **6** (2015), 167–195. <https://doi.org/10.3233/SW-140134>
3. F. M. Suchanek, G. Kasneci, G. Weikum, Yago: A core of semantic knowledge, in *Proceedings of the 16th International Conference on World Wide Web*, (2007), 697–706. <https://doi.org/10.1145/1242572.1242667>
4. G. A. Miller, WordNet: A lexical database for English, *Commun. ACM*, **38** (1995), 39–41. <https://doi.org/10.1145/219717.219748>
5. C. Xiong, R. Power, J. Callan, Explicit semantic ranking for academic search via knowledge graph embedding, in *Proceedings of the 26th International Conference on World Wide Web*, (2017), 1271–1279. <https://doi.org/10.1145/3038912.3052558>
6. Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, et al., An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (2017), 221–231. <https://doi.org/10.18653/v1/P17-1021>
7. F. Zhang, N. J. Yuan, D. Lian, X. Xie, W. Y. Ma, Collaborative knowledge base embedding for recommender systems, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2016), 353–362. <https://doi.org/10.1145/2939672.2939673>
8. L. Niu, C. Fu, Q. Yang, Z. Li, Z. Chen, Q. Liu, et al., Open-world knowledge graph completion with multiple interaction attention, *World Wide Web*, **24** (2021), 419–439. <https://doi.org/10.1007/s11280-020-00847-2>
9. D. Q. Nguyen, A survey of embedding models of entities and relationships for knowledge graph completion, in *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, (2020), 1–14. <http://doi.org/10.18653/v1/2020.textgraphs-1.1>
10. A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in *Advances in Neural Information Processing Systems*, **26** (2013), 2787–2795. Available from: <https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf>.
11. Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, (2014), 1112–1119. <https://doi.org/10.1609/aaai.v28i1.8870>
12. Y. Fang, X. Zhao, Z. Tan, S. Yang, W. Xiao, A revised translation-based method for knowledge graph representation, *J. Comput. Res. Dev.*, **55** (2018), 139–150. <https://doi.org/10.7544/issn1000-1239.2018.20160723>
13. Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, (2015), 2181–2187. <https://doi.org/10.1145/3132847.3133095>

14. G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (2015), 687–696. <https://doi.org/10.3115/v1/P15-1067>
15. B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in *Conference Track Proceedings of the 3rd International Conference on Learning Representations*, preprint, arXiv:1412.6575.
16. S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in *Advances in Neural Information Processing Systems*, preprint, arXiv:1802.04868.
17. I. Balazevic, C. Allen, T. M. 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), 5184–5193. <http://doi.org/10.18653/v1/D19-1522>
18. R. Socher, D. Chen, C. D. Manning, A. Y. Ng, Reasoning with neural tensor networks for knowledge base completion, in *Advances in Neural Information Processing Systems*, **26**, (2013), 926–934. Available from: <https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf>.
19. X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, et al., Knowledge vault: a web-scale approach to probabilistic knowledge fusion, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2014), 601–610. <https://doi.org/10.1145/2623330.2623623>
20. T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D knowledge graph embeddings, in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, (2018), 1811–1818. <https://doi.org/10.1609/aaai.v32i1.11573>
21. D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, D. Q. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (2018), 327–333. <http://doi.org/10.18653/v1/N18-2053>
22. C. Shang, Y. Tang, J. Huang, J. Bi, X. He, B. Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, (2019), 3060–3067. <https://doi.org/10.1609/aaai.v33i01.33013060>
23. S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, P. Talukdar, InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions, in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, (2020), 3009–3016. <https://doi.org/10.1609/aaai.v34i03.5694>
24. J. Huang, T. Zhang, J. Zhu, W. Yu, Y. Tang, Y. He, A deep embedding model for knowledge graph completion based on attention mechanism, *Neural Comput. Appl.*, **33** (2021), 9751–9760. <https://doi.org/10.1007/s00521-021-05742-z>

25. D. Jiang, R. Wang, J. Yang, L. Xue, Kernel multi-attention neural network for knowledge graph embedding, *Knowledge-Based Syst.*, **227** (2021), 107188. <https://doi.org/10.1016/j.knosys.2021.107188>
26. Z. Zhou, C. Wang, Y. Feng, D. Chen, JointE: Jointly utilizing 1D and 2D convolution for knowledge graph embedding, *Knowledge-Based Syst.*, **240** (2022), 108100. <https://doi.org/10.1016/j.knosys.2021.108100>
27. J. Feng, Q. Wei, J. Cui, J. Chen, Novel translation knowledge graph completion model based on 2D convolution, *Appl. Intell.*, **52** (2022), 3266–3275. <https://doi.org/10.1007/s10489-021-02438-8>
28. T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in *Proceedings of the 33rd International Conference on Machine Learning*, (2016), 2071–2080. Available from: <https://dl.acm.org/doi/10.5555/3045390.3045609>.
29. Z. Sun, Z. Deng, J. Nie, J. Tang, RotatE: Knowledge graph embedding by relational rotation in complex space, in *Proceedings of the 7th International Conference on Learning Representations*, (2019), 1–18. Available from: <https://www.researchgate.net/publication/331397037>.
30. S. Zhang, Y. Tay, L. Yao, Q. Liu, Quaternion knowledge graph embeddings, in *Advances in Neural Information Processing Systems*, preprint, arXiv:1904.10281.
31. Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: a survey of approaches and applications, *IEEE Transactions on Knowledge and Data Engineering*, **29** (2017), 2724–2743. <http://doi.org/10.1109/TKDE.2017.2754499>
32. Z. Liu, M. Sun, Y. Lin, R. Xie, Knowledge representation learning: a review, *J. Comput. Res. Dev.*, **53** (2016), 247–261. <https://doi.org/10.7544/ISSN1000-1239.2016.20160020>
33. A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: a comparative analysis, *ACM Trans. Knowl. Discovery Data*, **15** (2021), 1–49. <https://doi.org/10.1145/3424672>
34. F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, C. Li, Realistic re-evaluation of knowledge graph completion methods: an experimental study, in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, (2020), 1995–2010. <https://doi.org/10.1145/3318464.3380599>
35. 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), 57–66. <http://doi.org/10.18653/v1/W15-4007>
36. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 7132–7141. <https://doi.org/10.1109/CVPR.2018.00745>
37. V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in *Proceedings of the 27th International Conference on Machine Learning*, (2010), 807–814. Available from: <https://icml.cc/Conferences/2010/papers/432.pdf>.
38. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in *Conference Track Proceedings of the 3rd International Conference on Learning Representations*, preprint, arXiv:1412.6980.

39. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 2818–2826. <http://doi.org/10.1109/CVPR.2016.308>
40. X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in *2019 IEEE Conference on Computer Vision and Pattern Recognition*, (2019), 510–519. <https://doi.org/10.1109/CVPR.2019.00060>
41. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 11531–11539. <https://doi.org/10.1109/CVPR42600.2020.01155>
42. S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: Convolutional block attention module, in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, (2018), 3–19. https://doi.org/10.1007/978-3-030-01234-2_1



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)