Data & Knowledge Engineering 146 (2023) 102184

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

# Enhancing the convolution-based knowledge graph embeddings by increasing dimension-wise interactions[☆]

Fengyuan Lu [a,b], Jie Zhou [a,b], Xinli Huang [a,b,*]

[a] *School of Computer Science and Technology, East China Normal University, 3663 North Zhongshan Road, Shanghai, 200062, China*
[b] *NPPA Key Laboratory of Publishing Integration Development, ECNUP, China*

ARTICLE INFO

*Keywords:*
Knowledge graph embedding
Convolutional neural network
Link prediction

ABSTRACT

Knowledge graph embedding learns distributed low-dimensional representations for the elements in knowledge graphs, so that knowledge can be conveniently integrated into various tasks and smart systems. Recently, convolutional neural network has been introduced to embedding technique and obtained impressive achievements in link prediction task. ConvKB, a recently proposed method, captured the global dimension-wise interactions in facts with the convolutional filters. However, ConvKB failed to learn the local interactions between the entity and relation embedding. Moreover, rich interactions among feature maps are neglected in the existing convolutional embedding models. In this paper, based on ConvKB, we propose ConvD which models the local relationships in facts and integrates the cross-channel information based on the dimension-wise interactions to further improve the performance. From the experimental results, ConvD obtains scores that are 96% and 5% better than ConvKB on MRR and Hits@10 in the link prediction task. Furthermore, ConvD surpassed state-of-the-art baselines on WN18RR and achieved competitive results on FB15k-237 respectively.

## 1. Introduction

Knowledge graphs (KGs) are knowledge bases represented in graph structure, composed of entities and relations. Triples like (*head entity, relation, tail entity*) are usually used to represent facts in KG, e.g., (*Shanghai, cityOf, China*). Multiple KGs have been constructed, for instance, Freebase [1], DBpedia [2], and Wikidata [3]. KGs can be helpful to incorporate human knowledge into different downstream tasks like semantic parsing [4], question answering [5], and information extraction [6]. However, as symbolic representation, triples are difficult to operate [7]. Additionally, large-scale KGs are usually built through auto or semi-auto methods, which leads to the problem of incompleteness.

To handle the aforementioned issues, knowledge graph embedding (KGE) is proposed and applied to link prediction task which finds missing facts based on the existing facts in KG. KGE aims to map the entities and relations into low-dimensional vectors or matrices space, so that the observed triples can be easily operated to infer new facts. The embedding technique usually uses specific architecture to model the interactions between the embeddings and score the facts. Most conventional literature introduced methods based on translation distance [8–10] and semantic matching, using linear and bilinear models [11,12]. The structures of

☆ This work was supported in part by the National Key Research and Development Plan of China under Grant 2019YFB2012803, in part by the Key Project of Shanghai Science and Technology Innovation Action Plan, China under Grant 19DZ1100400 and Grant 18511103302, in part by the Key Program of Shanghai Artificial Intelligence Innovation Development, China Plan under Grant 2018-RGZN-02060, and in part by the Open Research Fund of NPPA Key Laboratory of Publishing Integration Development, ECNUP, China.

* Corresponding author at: School of Computer Science and Technology, East China Normal University, 3663 North Zhongshan Road, Shanghai, 200062, China.

*E-mail address:* xlhuang@cs.ecnu.edu.cn (X. Huang).

https://doi.org/10.1016/j.datak.2023.102184
Received 7 August 2022; Received in revised form 27 April 2023; Accepted 5 May 2023
Available online 11 May 2023
0169-023X/

these methods are relatively shallow. Neural networks have also been used to evaluate the plausibility of facts [13]. Compared with the translation and semantic matching-based methods, neural networks are deeper and considered to be more expressive. However, the large number of parameters makes these models prone to over-fitting and have lower computational efficiency.

Recently, convolutional neural networks (CNNs) are applied to do the KGE. Dettmers et al. [14] considered that the properties of parameter efficient and powerful expressiveness make CNNs have better scalability to large-scale KGs than other methods and proposed the first CNN-based embedding model ConvE. ConvKB [15] was proposed soon after ConvE and better results are achieved in their experiments. The former utilized 2D convolutions to catch local relationships between head entity and relation and the latter employed 1D on the global relationships in facts. Our work is mainly based on the ConvKB, for the translation characteristic captured with 1D is more effective, which is validated in the following sections. Another important reason is that ConvKB is less complex than ConvE, which leads to faster computation.

In this paper, we extend ConvKB and try to increase the dimension-wise interactions to enhance the performance of the model. We propose ConvD, which models the local, global relationship in the facts and integrates the cross-channel information. ConvKB only caught and heavily count on the dimension-wise interactions in the convolutional filters to learn the global relationship in the triples. In the previous classical translation-based methods: TransH [16] and TransR [10], relation-specific entity embedding is introduced to improve the performance. We inferred that such preprocessing can also be helpful to ConvKB. As a result, we model the local relationship between the entity and relation in advance of the convolutional operation. In addition, we pay attention to the cross-channel information after the convolution. To the best of our knowledge, catching interactions among feature maps is popular in computer vision but has never been used in CNN-based KGE methods. We adjusted and applied the methods in computer vision to our model for integration of all feature maps, expecting to improve the experimental results. As we encode the mentioned information based on dimension-wise interactions between the embeddings so that it can be interpreted that we improve ConvKB with the additional dimension-wise interactions. The structure of ConvD is concise and defined by a Hadamard product layer, a single convolutional layer, and a set of dimension-wise fully connected layers. The source code of the paper has been publicly available at https://github.com/lab606/ConvD.

In summary, our contributions are listed as follows:

- We propose ConvD, a method that boosts the performance of ConvKB through two crucial mechanisms —catching the local relationship and integrating the cross-channel information.
- We set up a series of experiments to evaluate ConvD. The demonstrated experimental results obtain advanced performance on most metrics of WN18RR [14] and FB15k-237 [17] and no unusual score distribution is detected on the two datasets.

## 2. Background

**Definition 2.1** (*Knowledge Graph (KG) $\mathcal{G}$*)**.** The definition of KG includes three parts, the set of entities $\mathcal{E}$, the set of relations $\mathcal{R}$ and the set of facts $\mathcal{F}$, i.e., $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$. Some KGs may contain auxiliary, like textual information and attributes, but here we only consider the universal setting.

**Definition 2.2** (*Knowledge Graph Embedding*)**.** To learn the KG embeddings, a KG embedding model should select its representation space and define the encoding model and scoring function. The encoding model refers to different model architectures designed to encode the interactions between embeddings. Whether the fact is correct or not depends on the plausibility measured by the scoring function. To learn the distributed representations of the components is to maximize the entire plausibility of facts in KG, and then the representations can be used to infer the new facts in the link prediction task.

## 3. Related work

KG embedding has been an active research area in the past couple of years. Various types of representation learning methods were proposed to embed multi-relational knowledge into low-dimensional representations of entities and relations, including translation-based models, matrix/tensor factorization models, and deep learning models. Translation-based models usually treat relation as a translation from head entity to tail entity. TransE applies well to 1-to −1 relations but has issues for N-to −1, 1-to-N, and N-to-N relations. TransH, and TransR, the extensions of TransE, further introduced relation-specific hyperplanes [10] and space [9] to handle these complex relations. Other translation-based models utilize different representation spaces, such as Gaussian space [18] and manifold [19]. RotatE [20] got the intuition from the Euler's identity and regarded the relation as a rotation from head entity to tail entity in the complex vector space. The tensor factorization methods represent KG as a tensor and consider identifying triples as a process of tensor decomposition. Much like with translation-based models, these approaches usually avoid shared parameters, running back-propagation directly on the embeddings. RESCAL [21] modeled the interaction between the entity pair in a bilinear formulation. DistMult focused on the symmetric relation and restrict the bilinear formulation to be diagonal. Based on DistMult, ComplEx [22] introduced complex embedding, so that asymmetric relation can be learned as well. In the early stage of KGE research, there were also a few works based on neural network. Neural Tensor Network (NTN) [13] utilized a neural network to score the facts, which modeled the relationship between the entities with a relation-specific tensor. Multi-Layer Perception (MLP) [23] is also used to encode the facts. Entities and relations are concatenated and input into the fully-connected layers to encode the semantic matching. Although these models were considered to have a strong ability to capture features, the huge number of parameters and overfitting problem prevented them from achieving desirable results.

**Table 1**
Scoring functions of introduced KGE models.

| Model | Scoring function $f_r(h,t)$ |
|---|---|
| TransE | $\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2$ |
| DistMult | $\langle \mathbf{h},\mathbf{r},\mathbf{t}\rangle$ |
| ComplEx | $Re(\langle \mathbf{h},\mathbf{r},\mathbf{t}\rangle)$ |
| ConvKB | $concat(f([\mathbf{h},\mathbf{r},\mathbf{t}]\star\omega))\cdot\mathbf{w}$ |
| ConvE | $f(vec(f([\bar{\mathbf{h}};\bar{\mathbf{r}}]\star\omega))\mathbf{W})\cdot\mathbf{t}$ |
| InteractE | $g(vec(f(\phi(\mathbf{P}_k)*\omega))W)\cdot\mathbf{t}$ |

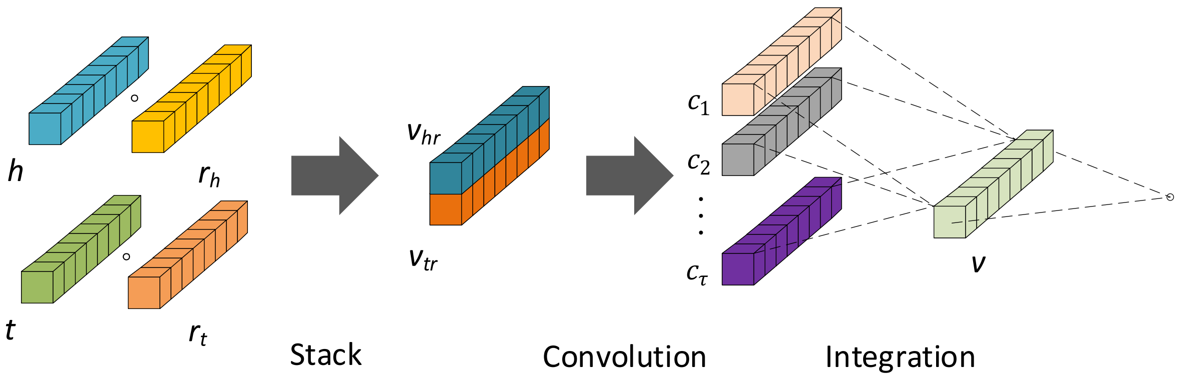


**Fig. 1.** In ConvD, the entity embeddings $(\mathbf{h},\mathbf{t})$ is first Hadamard producted with the corresponding relation embedding $(\mathbf{r}_h,\mathbf{r}_t)$; the generated embeddings $(\mathbf{v}_{hr},\mathbf{v}_{tr})$ are stacked as the input matrix of the convolutional layer; entry-wise linear layers are used to integrate the interactions of the feature maps $\mathbf{c}_i$.

However, as the research continues and various schemes to avoid overfitting are proposed. Deep learning models are back in focus. Convolutional Neural Network (CNN) has achieved many remarkable achievements in the field of computer vision, showing its parameter efficiency and powerful expressiveness. As a result, it made researchers think that CNNs have better scalability to large-scale KGs than the above-mentioned work. Recently, multiple CNN-based methods have been proposed. They mainly stack or reshape the embeddings of the entities and relations first. Filters are then applied to extract various features from the input matrix. ConvE reshaped and concatenated the embeddings of head entity and relation as the input matrix to the convolution layer. $3\times 3$ filters are employed over the matrix to compute the map features which are then compared with all tail entities. ConvKB represented each triple as the input matrix and catches the global relationships at the same dimension with $1\times 3$ filters. In comparison to ConvE, ConvKB retained the translation characteristic. Shang et al. [24] proposed Conv-TransE, which captured the translation characteristic in the formulation of ConvE. Besides, they proposed a weighted graph convolutional model SACN combined with Conv-TransE. InteractE [25] is constructed based on ConvE, which increased the possible interactions between embeddings by permuting the features, reshaping the embedding in a checked form, and using the depth-wise circular convolution. The results proved that increasing interactions can be helpful to promote performance. ReInceptionE [26] gathered the relational local and attention-based global entity information to improve the embedding learning in the Inception network. Scoring functions $f_r(h,t)$ of some representative and latest KG embedding models are listed in Table 1. More existing KG embedding models are available in recent surveys [7,27]. In ComplEx $\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^d$ and for the other methods they are real vectors $(\mathbb{R}^d)$. $\star$ denotes convolution, $\cdot$ denotes a dot product and $*$ denotes depth-wise circular convolution. In this paper, we utilize local relationship and cross-channel information to construct our novel CNN-based model, which can increase dimension-wise interactions.

## 4. Proposed ConvD

In this section, we provide a detailed introduction to the elements of ConvD. The overall architecture of ConvD is illustrated in Fig. 1. In ConvD, all the entities and relations are embedded in a real value $d$-dimensional vector space $\mathbb{R}^d$.

### 4.1. Catch the local relationship

Relations are usually regarded as the transition or bridge between the head and tail entity. We consider that, in a triple, relation can share the equal status with entity and the local relationship between them should not be completely abandoned. For the plausibility of the local relationship is a necessary condition for the correct triples and a sufficient condition for corrupted triples. For instance, given the triple *(Shanghai, Capital of, China)*, we can tell that it is a negative fact only depending on the (*Shanghai, Capital of*). As a result, we treat the triple from part to integer. Given a fact $(h,r,t)$, we first separate it into two parts. They are respectively composed of $h,r$, and $t,r$. The reason why we put the relation into two isolated pairs is that we think the relation should have distinct relevance with the head and tail entity if the triple is observed locally. For example, *(Shanghai, cityOf, China)* is divided into *(Shanghai, cityOf)* and *(cityOf, China)*. *city of* for *Shanghai* and *cityOf* for *China* is semantically different. Within the pair, we try to model the local interactions. Particularly, we represent $h$ and $t$ with the vectors $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ respectively. The relation $r$ is associated with the embedding pair $\mathbf{r} = (\mathbf{r}_h, \mathbf{r}_t)$. We catch the dimension-wise interactions by computing the semantic similarity between the entity and relation embeddings:

$$\begin{aligned} \mathbf{v}_{hr} &= \mathbf{h} \circ \mathbf{r}_h \\ \mathbf{v}_{tr} &= \mathbf{t} \circ \mathbf{r}_t, \end{aligned} \tag{1}$$

where $\mathbf{v}_{hr}$ and $\mathbf{v}_{tr}$ are the generated feature vectors of the local relationship in entity-relation pairs.

### 4.2. Catch the global relationship

In the previous subsection, the local relationship is extracted from the entity-relation pairs via Hadamard product. Even if the separator pairs are both correct, they are not necessarily positive after being combined. Therefore, global relationship between the generated feature vectors is further studied. Particularly, we use convolution filters to catch the dimension-wise interactions between the local feature vectors. The compositive vectors are stacked as the input matrix $\mathbf{A} = [\mathbf{v}_{hr}, \mathbf{v}_{tr}]$, where $\mathbf{A}_{i,:} \in \mathbb{R}^{1\times 2}$ denotes the $i$th row of the matrix. In the convolution layer, we apply width 1 filters over the stacked input. More information about why the size of filter is fixed at 2 is presented in Section 4.4. Convolution filter $\boldsymbol{\omega} \in \mathbb{R}^{1\times 2}$ is operated on each row:

$$c_i = g(\boldsymbol{\omega} \cdot \mathbf{A}_{i,:} + b), \tag{2}$$

to learn the global relationship and generate a feature map $\mathbf{c} = [c_1, \dots, c_d]$. As these two local pairs also have different combination mode, such as 1-1, 1-N, N-N and symmetry. With multiple filters, we can learn various relation patterns. A feature map tensor $\mathbf{C} \in \mathbb{R}^{\tau\times d\times 1}$ is obtained after the convolution, where $\tau$ is the number of filters.

### 4.3. Integrate the cross channel information

In ConvKB, the map tensor is directly reshaped into a flat vector which is then computed with the weight vector via the inner product to evaluate the fact. The strategy is simple and efficient but still can be improved. In the previous phase, global relationship is learned and various feature maps are generated. However, relationship patterns in fact are limited. For example, a filter for N-N can hardly extract valid features from a symmetric triple. The corresponding feature map is useless or negative for evaluation. On the other hand, valuable information among the channels should not be neglected, depending on the application of the multi-layer CNN models in computer vision. As a result, it occurred to us to integrate the cross-channel information. At this step, eliminating the influence of unrelated feature maps and learning more semantic features among valuable channels is expected to be put into effect. For the operation concentrated on the entry-wise features in different channels, the $1 \times 1$ filter is intuitively to be considered.

In image processing, $1 \times 1$ convolutional filter is widely used, such as in NIN [28], Inception [29] and ResNet [30]. The specific filter can be used to deepen and widen the network structure, control the dimension with fewer parameters and integrate the pixels in different channels. As far as ConvD is concerned, $1 \times 1$ filter gives channels different weights to determine their importance from the perspective of interaction between channels. Based on the analysis above, a convolution layer with a single $1 \times 1$ filter can be applied to integrate the cross-channel information. To further improve the performance of the model, we upgrade the iterative convolution on each dimension to the full connection operation. It is a set of dimension-wise linear transform layers which treats the columns across the channel as input so that the manipulation on various dimensions is independent. Additionally, compared to the models in computer vision, ours is still shallow and needs more expressiveness. The map tensor $\mathbf{C} \in \mathbb{R}^{\tau\times d\times 1}$ is firstly reshaped into matrix $\mathbf{M} \in \mathbb{R}^{d\times\tau}$, then the matrix is further split to $\mathbf{m}_{1\le i\le d} \in \mathbb{R}^{\tau}$ and the integration formulation is defined as:

$$v_i = g(\mathbf{m}_i^{\top}\mathbf{w}_i + b_i), 1 \le i \le d \tag{3}$$

where $\mathbf{w}_i \in \mathbb{R}^{\tau}$ is the weight matrix, $b$ denotes bias term and $g$ is the activation function, $v_i$ is the $i$th entry of the final embedding. Formally, the final scoring function is defined as:

$$f_r(h,t) = concat(v_1, v_2, \dots, v_d) \cdot \mathbf{w} \tag{4}$$

where $\mathbf{w} \in \mathbb{R} \in d$ is the weight vector. The scoring function can also be interpreted as the plausibility of the triple is measured by scoring each interaction at the same dimension entry and calculating their total contributions.

Eventually, we define logistic loss for KG embedding as:

$$\begin{aligned} \mathcal{L} = \sum_{(h,r,t)\in\mathcal{F}\cup\mathcal{F}'} & log(1 + \exp(-y_{hrt} \cdot f_r(h,t))) \\ & + \frac{\lambda}{2} \parallel \Theta \parallel_2^2 \end{aligned} \tag{5}$$

Here, $\mathcal{F}$ and $\mathcal{F}'$ denote the positive and negative examples respectively, where $y_{hrt}$ is the label of example, 1 for the positive and $-1$ for the negative. The hyper-parameter $\lambda$ is used on the $L2$ regularization of model parameters $\Theta$.

### 4.4. Translation characteristic analysis

Translation characteristic, proposed in ConvKB, is an intuitive idea obtained from TransE and applied in convolution operation. Here, we try to carry out some analysis to explain why the characteristic is valid based on the definition of the interactions in

**Table 2**
Statistic of the datasets.

| Dataset | $\mathcal{E}$ | $\mathcal{R}$ | #Triples | | |
|---|---|---|---|---|---|
| | | | Train | Valid | Test |
| FB15k-237 | 14,541 | 237 | 272,115 | 17,535 | 20,466 |
| WN18RR | 40,943 | 11 | 86,835 | 3,034 | 3,134 |

convolutional filters provided by Vashishth et al. [25]. Let $M_k$ denote the sub matrix of $\mathbf{A}$ covered with $2 \times k$ filter, and $x, y \in M_k$ denote variant elements in $\mathbf{v}_h r$ or $\mathbf{v}_t r$. Such interaction is defined as triple $(x, y, M_k)$ and the number is the cardinality of the triples. Moreover, interactions can be classified into heterogeneous and homogeneous, depending on whether $x$ and $y$ come from $\mathbf{v}_{hr}$ and $\mathbf{v}_{tr}$ respectively or not. More heterogeneous and less homogeneous can bring about better performance, which is verified in InteractE. Let $\mathcal{N}_{heter}$ and $\mathcal{N}_{homo}$ denote the number of heterogeneous and homogeneous respectively. We can define the proportion of the two kinds of interactions in width $k$ filter:

$$\begin{cases} \dfrac{\mathcal{N}_{homo}}{\mathcal{N}_{heter}} = \dfrac{\binom{k}{2}}{k^2} = \dfrac{1}{2} - \dfrac{1}{2k}, k > 1, k \in Z^+, \\ \dfrac{\mathcal{N}_{homo}}{\mathcal{N}_{heter}} = 0, k = 1. \end{cases} \tag{6}$$

From the equation, as the width of filter grows, the proportion $\mathcal{N}_{homo}$ occupies increases. In the case of $k = 1$, no $\mathcal{N}_{homo}$ is introduced in the filter. It can be inferred that catching the translation characteristic is more practical to improve the performance. Corresponding experiments is provided in Section 6.5.

## 5. Experiments setup

### 5.1. Datasets

We evaluate ConvD by executing a link prediction task on the two most widely used datasets: FB15k-237 [17] and WN18RR [14]. FB15k-237 is the subset of FB15k, which was generated by Toutanova et al. [17] by deleting the facts with inverse relation. Dettmers et al. [14] followed the idea and removed the inverse relations in WN18 to create WN18RR. Furthermore, they investigated the test leakage issue in WN18 and FB15k. Remarkable results on these two datasets can be achieved by using a method designed to only model inverse relations. As a result, WN18RR and FB15k-237 are more recommended for evaluation. Corresponding statistics of the datasets are presented in Table 2.

### 5.2. Evaluation protocol

Link prediction, a type of KG completion task, compares the entity with a specific relation to other entities and ranks the candidate facts depending on their score. Following Bordes et al. [8], "Filter" setting is used, i.e., when evaluating on the test set, the valid triples are excluded, which are produced by corrupting the entities. Several standard evaluation metrics are employed: Mean Rank (MR), Mean Reciprocal Rank (MRR), Hits@1, Hits@3, and Hits@10. Lower MR, higher MRR, and Hits@N imply the better performance of the model. Recently, Sun et al. [31] found unusual score distributions in recent neural network (NN)-based models, where some negative triples have the same scores as true triples. They counted an instance from the FB15k-237 dataset, out of 14,541 negatively sampled triples, 8520 had the same score as the real triples. This unusual score distribution stems from the misuse of the evaluation protocol, which can lead to inflated results.

As a result, we employed the recommended RANDOM evaluation rule. Among the multiple triples scored equally by the model, the true triple is placed at the random position in the candidate set $\Gamma$. Then the triples in $\Gamma$ are sorted stably and the metrics are generated statistically based on the order of the triples.

### 5.3. Training protocol

Adding negative samples in the training procedure is considered to be helpful to enhance the performance. We generate the negative samples, following the Bernoulli distribution [10]. To learn the embeddings and model parameters, we select the hyper-parameters via grid search depending on the Hits@10. The initial learning rate is selected from $\{1e^{-4}, 8e^{-5}\}$ and adam optimization is adopted. The range of the embedding size is $\{200, 250, 300, 350, 400\}$ for WN18RR and $\{600, 700, 800\}$ for FB15k-237, the filters is $\{16, 24, 32, 48\}$, the dropout rate is $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, the $L_2$ regularization is $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Eventually, we set the embedding size to 350 for WN18RR and 800 for FB15k-237, the learning rate to $8e^{-5}$, the number of filters to 48 for WN18RR and 16 for FB15k-237, the dropout rate to 0.2, the $L2$ regularization to 0.2. The best result is obtained after running 1000 epochs on both datasets.

## 6. Results

In this section, according to the experimental results, we want to demonstrate: (1) the performance of ConvD compared with other existing methods, (2) different results from the same model, (3) whether ConvD suffers from unusual score distribution, (4)

**Table 3**
Results of different work which are obtained by executing the link prediction task on datasets WN18RR and FB15k-237. Results of TransE derived from [15]. [∗] denotes the source is [14]. Work with superscript 1 means the re-evaluated results from [31], while with superscript 2 represents the original results. Others are taken from the corresponding paper. The top performance is marked in **bold** and the second-best in underline.

| Models | WN18RR | | | | | FB15k-237 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MR | MRR | @1 | @3 | @10 | MR | MRR | @1 | @3 | @10 |
| TransE [8] | 3384 | .226 | – | – | .501 | 347 | .294 | – | – | .465 |
| DistMult [11][∗] | 5110 | .430 | .390 | .440 | .490 | 254 | .241 | .155 | .263 | .419 |
| ComplEx [22][∗] | 5261 | .440 | .410 | .460 | .510 | 339 | .247 | .158 | .275 | .428 |
| ConvE [14] | 4187 | .430 | .400 | .440 | .520 | 244 | .325 | .237 | .356 | .501 |
| RotatE [20] | 3340 | .476 | .428 | .492 | .571 | 177 | .338 | .241 | .375 | .533 |
| ConvR [32] | – | .475 | .443 | .489 | .537 | – | .350 | .261 | .385 | .528 |
| SACN [24] | – | .470 | .430 | .480 | .540 | – | .350 | .260 | **.390** | **.540** |
| QuatE [33] | 3472 | .481 | .436 | .500 | .564 | 176 | .311 | .221 | .342 | .495 |
| InteractE [25] | 5202 | .463 | .430 | – | .528 | 172 | **.354** | **.263** | – | .535 |
| ReInceptionE [26] | 1894 | .483 | – | – | **.582** | 173 | .349 | – | – | .528 |
| ComplEx-N3-RP [34] | – | .480 | – | – | .570 | – | .370 | – | – | .560 |
| ConvKB [15]$^{1}$ | 3433 | .249 | – | – | .524 | 309 | .243 | – | – | .421 |
| CapsE [35]$^{1}$ | **718** | .415 | – | – | .559 | 403 | .150 | – | – | .356 |
| KBGAT [36]$^{1}$ | 1921 | .412 | – | – | .554 | 270 | .157 | – | – | .331 |
| ConvD | 2856 | **.489** | **.445** | **.506** | .571 | **156** | .342 | .248 | .377 | .531 |
| ConvKB [15]$^{2}$ | 2554 | .248 | – | – | .525 | 257 | .396 | – | – | .517 |
| CapsE [35]$^{2}$ | 719 | .415 | – | – | .560 | 303 | .523 | – | – | .593 |
| KBGAT [36]$^{2}$ | 1940 | .440 | .361 | .483 | .581 | 210 | .518 | .460 | .540 | .626 |

effect of different information on ConvD, (5) the validation of translation characteristic (6) the performance of ConvD when faced with different relation types.

### *6.1. Performance comparison*

As mentioned earlier, Sun et al. [31] find out the evaluation issue in the NN-based models, including ConvKB [15], CapsE [35] and KBGAT [36]. ConvKB and CapsE suffered from the unusual score distribution. Test leakage is found in KBGAT. In the re-evaluation experiment, the distribution was corrected and the leakage was eliminated. Yet, on a part of metrics, the obtained results are very different from those in the original paper. For fairness and clarity, we listed both the reported results from two sources and those with large differences were excluded from the comparison. ComplEx-N3-RP was also not compared to the others because its focus is on optimization methods rather than proposing a new model.

From the results observed in Table 3, it can be found that ConvD has achieved excellent performance, compared with other models, on the datasets WN18RR and FB15k-237. Especially on WN18RR, ConvD has achieved the best MRR, Hits@1, Hits@3, and the second-best Hits@10 score. On FB15k-237, ConvD achieves the best MR and the second best Hits@3 score, the other metrics also get competitive outcomes. As an extension of ConvKB, ConvD increased the Hit@1 score by $96.3\%((0.489-0.249)/0.249)$ and Hits@10 score by $5\%((0.573-0.534)/0.534)$ on WN18RR. For results on the FB15k-237 of ConvKB became controversial after the re-evaluation experiment, more detailed comparisons with more evaluation protocols and analysis are demanded and provided in the next subsection. The above comparison shows that the design of ConvD is valid and efficient. Catching the translation characteristic of triples from the local to the whole and cross-channel information is helpful to knowledge representation. At the same time, the best results on the two datasets are both obtained by the CNN-based methods, indicating the effectiveness of the application of CNN in KGE.

By comparison, the performance of ConvD on WN18RR is not good as that on FB15k-237. Based on the statistic of the datasets, FB15k-237 owns a higher average degree. Every entity in that is connected with more other entities, so it has a rich neighborhood. But ConvD was only concerned with the internal information of the facts and failed to take advantage of the external neighborhood information. Based on CNN-based decoders, just like ConvD, ReInceptionE, and SACN introduced GCN-based encoders to catch the neighborhood information or called structure information to learn the entity embeddings. As a result, they achieved superb results on FB15k-237. Compared with ConvD, InteractE achieves better performance on FB15k-237 with more complicated mechanisms and more interactions among different embedding entries. ConvR also treats relations as convolution filters and generates interactions on various entries with entities. It infers that such interaction is more suitable for datasets with high average degrees, but poor at handling the sparse dataset. Unlike the other embedding techniques, ComplEx-N3-RP is not a newly-proposed model but uses N3 regularizer and reciprocals to help ComplEx reach the top of FB15k-237 and greatly improve the performance on WN18RR. Whether these methods can be applied to other models is worth studying in the future.

### *6.2. Different results from the same model*

Experimental results of different models are gathered in Table 3 for the comparison with ConvD. Notably, we have retained the results of ConvKB, CapsE, and KBGAT from the original and the re-evaluation experiments, because the results vary in numerical

**Table 4**
Various results of the mentioned models are collected from different sources. [⋆] indicates the data are derived from [15]. [∗] denotes the source is [14]. Models with superscript 1 means the results are taken from [37], while with superscript 2 represents those are from [31].

| Models | WN18RR | | | | | FB15k-237 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MR | MRR | @1 | @3 | @10 | MR | MRR | @1 | @3 | @10 |
| TransE [8][⋆] | 3384 | .226 | – | – | .501 | 347 | .294 | – | – | .465 |
| TransE [8][1] | 3936 | .206 | .028 | – | .495 | 209 | .31 | .217 | – | .497 |
| DistMult [11][∗] | 5110 | .430 | .390 | .440 | .490 | 254 | .241 | .155 | .263 | .419 |
| DistMult [11][1] | 5913 | .433 | .397 | – | .502 | 199 | .313 | .224 | – | .490 |
| ComplEx [22][∗] | 5261 | .440 | .410 | .460 | .510 | 339 | .247 | .158 | .275 | .428 |
| ComplEx [22][1] | 4907 | .458 | .426 | – | .521 | 202 | .349 | .257 | – | .530 |
| ConvE [14] | 4187 | .430 | .400 | .440 | .520 | 244 | .325 | .237 | .356 | .501 |
| ConvE [14][1] | 4944 | .427 | .390 | – | .508 | 281 | .305 | .219 | – | .476 |
| ConvKB [15] | 2554 | .248 | – | – | .525 | 257 | .396 | – | – | .517 |
| ConvKB-TOP [15][2] | 1696 | .251 | – | – | .529 | 246 | .407 | – | – | .527 |
| ConvKB-RANDOM [15][2] | 3433 | .249 | – | – | .524 | 309 | .243 | – | – | .421 |
| ConvKB-BOTTOM [15][2] | 5168 | .164 | – | – | .516 | 373 | .130 | – | – | .383 |
| ConvKB-AVERAGE [15][1] | 3429 | .249 | 0.056 | – | .525 | 309 | .230 | .140 | – | .415 |
| CapsE [35] | 719 | .415 | – | – | .560 | 303 | .523 | – | – | .593 |
| CapsE-TOP [35][2] | 718 | .415 | – | – | .559 | 305 | .511 | – | – | .586 |
| CapsE-RANDOM [35][2] | 720 | .415 | – | – | .559 | 403 | .150 | – | – | .356 |
| CapsE-BOTTOM [35][2] | 719 | .323 | – | – | .555 | 502 | .134 | – | – | .297 |
| CapsE-AVERAGE [35][1] | 720 | .415 | .337 | – | .560 | 405 | .160 | .073 | – | .356 |

value. Some other work reproduced the experiments in previous studies for comparison or analysis, in which the results were also found to be different from each other or the original. Faced with this potentially puzzling phenomenon, comparisons were made and results with a gap of more than 10% on a certain indicator were kept in Table 4. We try to briefly discuss what leads to the variance across different publications and how it affects ConvD.

After checking the details in the experiment section of related papers, the cause of the differences can be generally identified as the following factors, hyperparameter settings, regularization, and evaluation protocols. As Kadlec et al. [38] showed, hyperparameters usually have a decisive influence on the performance of embedding model. Results of TransE and DistMult, depending on the description in corresponding sources, are obtained by using different hyperparameter combinations. Because these traditional models were proposed before WN18RR and FB15k-237 existed, later researchers were left to find the best combinations on these data sets by themselves. ConvE faces a similar situation, the acceptable choices of hyperparameters are provided but the best combination is not clearly pointed out. For ComplEx, Lacroix et al. [34] proposed a new version with the norm-based N3 regularizer. If the referenced version is different, the displayed results will also be changed. The new version has shown regularization can be the future optimization direction of ConvD in the future. The multiple distinct results of ConvKB and CapsE, mainly on FB15k-237, are obtained with different evaluation metrics. But the root is the unusual distribution, many different triples are assigned the identical score. If the valid triple is placed top, bottom, random, or average position among all the same scores, different results are yielded. Since ConvD is an extension of ConvKB, it is necessary to check whether the same occurs on ConvD.

### 6.3. Different evaluation protocols on ConvD

Comparing the experimental results of MRR given in the original paper of ConvKB and ConvE, it can be found that the former is $21.8\%((0.396 - 0.325)/0.325)$ higher than the latter on the FB15k-237, but $42.3\%((0.430 - 0.248)/0.430)$ lower on WN18RR. For this relative inconsistency, Sun et al. raised questions and launched a research. The research exposed that ConvKB has the unusual score distribution issue which is caused by giving the positive and negative triples the same score. The general protocol makes the positive triples ahead of the other negative ones so that the evaluation results look great. However, the model actually failed to distinguish the triples. As said in the previous section, to check whether ConvD will bring out such distribution, we followed the re-evaluation procedure to apply two additional protocols, i.e., TOP and BOTTOM on ConvD. In TOP and BOTTOM, the positive triple is respectively inserted at the beginning and end of candidate set $\Gamma$. The results are listed in Table 5, from which we can see that ConvD obtained close performance under different evaluation protocols. Compared with ConvKB, ConvD achieved better results on most metrics, except the MRR score under TOP protocol. As Sun et al. recommended using RANDOM protocol for evaluation. Hence, combined with the results in Table 3, the performance of ConvD is better than ConvKB on both WN18RR and FB15k-237. More importantly, it is proved that ConvD avoids the unusual distribution and represents the facts distinctively.

### 6.4. Impact of different information

To find out the impact of local relationship and cross-channel information on the performance of the model, we design ConvKB as the baseline model and extend it with different tactics to execute the link prediction task on WN18RR. From Table 6, we can see that the performance of the model is significantly improved by catching the local relationships, which validates that the local

**Table 5**
Effect of different evaluation protocols on ConvD and ConvKB on FB15k-237 data set. For TOP and BOTTOM, changes in the performance under RANDOM protocol are also marked. The results of ConvKB are derived from [31].

| | TOP | | | RANDOM | | | BOTTOM | | |
|---|---|---|---|---|---|---|---|---|---|
| | MRR | MR | Hits@10 | MRR | MR | Hits@10 | MRR | MR | Hits@10 |
| ConvD | .342 | **157** (+1) | **.532** (+.001) | .342 | **156** | .531 | **.340** (−.002) | **156** | **.532** (+.001) |
| ConvKB | **.407** (+164) | 246 (−63) | .527 (+.106) | .243 | 309 | .421 | .130 (−.113) | 373 (+64) | .383 (−.038) |

**Table 6**
Effects of different information to the KGE task on WN18RR.

| Models | MR | MRR | Hits@1 | Hits@3 | Hits@10 |
|---|---|---|---|---|---|
| ConvKB | **2333** | .348 | .260 | .394 | .514 |
| ConvKB+Local Relationships | 2692 | .468 | .423 | .483 | .556 |
| ConvKB+Cross Channel Information | 2462 | .398 | .326 | .436 | .527 |
| ConvD | 2856 | **.489** | **.445** | **.506** | **.573** |

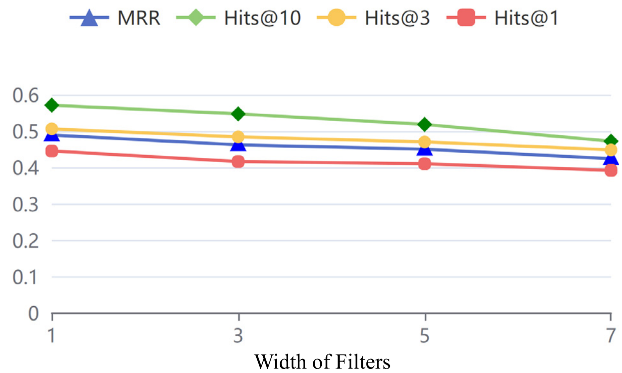


**Fig. 2.** Performance on WN18RR with different size of filters.

relationship between the head/tail entity and relation is helpful to measure the plausibility of the triple. The extension with cross-channel information increases relatively small, for we actually integrate the feature map but do not introduce more new valuable information. When the two schemes are both adopted, it comes to ConvD with impressive expressiveness. It is worthy noticing that ConvKB obtains the best MR score. We consider that a part of triples may be sensitive to the global relationship. We corrupt the integrity of the facts when separating them into two parts and corrupt the global relationship.

### 6.5. Impact of different size of filters

To validate the proposition in Section 4.4, we analyzed the results obtained by using different sizes of filters, including $1 \times 2, 3 \times 2, 5 \times 2$, and $7 \times 2$, on WN18RR. As Fig. 2 summarized, it can be observed that increasing the size of filters makes results worse on all the metrics, which confirms the proposition that larger filters can introduce more homogeneous interactions, leading to worse performance. To some extent, it also explains why translation characteristic is valid in CNN-based KGE.

### 6.6. Evaluation on different relation types

To study the performance of ConvD on the complex relations, we select ConvE and ReInception as the baseline and compare the Hits@10 metric on WB18RR and FB15k-237. We follow [8] to divide the relations into four categories: one-to-one (1-1), one-to-many (1-N), many-to-many (N-N,) and many-to-one (N-1), depending on the average head entities each tail entity owns and vice versa. Experimental results are listed in Table 7, from which we can find that ConvD outperforms ConvE on all types. Compared to ReInceptionE, ConvD achieves better performance on one-to-one relation, one-to-many relation in FB15k-237 and many-to-one relation in WN18RR. For the many-to-many relation, they get close results. Generally speaking, more interactions can help the model capture the complex relation more effectively. For ReInceptionE takes the structure information, the factors affecting the link prediction results on complex relation are worthy of further study.

## 7. Conclusion

In this paper, we proposed ConvD, a CNN-based KGE model extended from ConvKB, which utilizes the additional dimension-wise to catch the local relationship and integrate the cross-channel information. Experimental results show that ConvD achieves significant

**Table 7**
Hits@10 for each type of relation on the WN18RR and FB15k-237. Results of ConvE and ReInceptionE are taken from Xie et al. [26]. Four categories of relations are included: one-to-one (1-1), one-to-many (1-N), many-to-one (N-1) and many-to-many (N-N).

| | Models | Head prediction | | | | Tail prediction | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1–1 | 1-N | N-1 | N-N | 1–1 | 1-N | N-1 | N-N |
| FB15k-237 | ConvE | 0.303 | 0.590 | 0.137 | 0.400 | 0.272 | 0.845 | 0.088 | 0.545 |
| | ReInceptionE | 0.609 | 0.651 | **0.185** | 0.473 | 0.594 | 0.872 | **0.149** | 0.603 |
| | ConvD | **0.620** | **0.660** | 0.174 | **0.474** | **0.600** | **0.874** | 0.142 | **0.608** |
| WN18RR | ConvE | 0.975 | 0.414 | 0.110 | 0.950 | 0.975 | 0.303 | 0.153 | 0.949 |
| | ReInceptionE | **0.976** | **0.586** | 0.152 | **0.961** | **0.976** | **0.494** | 0.272 | **0.958** |
| | ConvD | **0.976** | 0.486 | **0.233** | 0.960 | **0.976** | 0.408 | **0.282** | 0.954 |

improvement and state-of-the-art performance on WN18RR and FB15k-237. Moreover, ConvD avoids the unusual score distribution found in NN-based models. We also analyze the translation characteristics and give the corresponding validation. Through the ablation experiment, we demonstrate the effectiveness of each scheme in ConvD. In the future, an interesting direction is to enhance ConvE by increasing the dimension-wise interactions.

## CRediT authorship contribution statement

**Fengyuan Lu:** Conceptualization, Methodology, Software, Writing – original draft. **Jie Zhou:** Writing – review & editing. **Xinli Huang:** Project administration, Writing – review & editing, Resources.

## 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.

## References

[1] K. Bollacker, C. Evans, P. 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, pp. 1247–1250.
[2] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semant. Web 6 (2) (2015) 167–195.
[3] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Commun. ACM 57 (10) (2014) 78–85.
[4] J. Berant, A. Chou, R. Frostig, P. Liang, Semantic parsing on freebase from question-answer pairs, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1533–1544.
[5] A. Bordes, J. Weston, N. Usunier, Open question answering with weakly supervised embedding models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2014, pp. 165–180.
[6] J. Daiber, M. Jakob, C. Hokamp, P.N. Mendes, Improving efficiency and accuracy in multilingual entity extraction, in: Proceedings of the 9th International Conference on Semantic Systems, 2013, pp. 121–124.
[7] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (12) (2017) 2724–2743.
[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 2787–2795.
[9] 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), Association for Computational Linguistics, 2015, pp. 687–696.
[10] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29. No. 1, 2015.
[11] B. Yang, W. tau Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: 3rd International Conference on Learning Representations, 2015.
[12] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: International Conference on Machine Learning, PMLR, 2016, pp. 2071–2080.
[13] R. Socher, D. Chen, C.D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 26, Curran Associates, Inc., 2013, pp. 926–934.
[14] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D knowledge graph embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1, 2018.
[15] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen, D. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: NAACL HLT 2018: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies-Proceedings of the Conference, Association for Computational Linguistics, 2018, pp. 327–333.
[16] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 28, No. 1, 2014.
[17] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1499–1509.
[18] S. He, K. Liu, G. Ji, J. Zhao, Learning to represent knowledge graphs with Gaussian embedding, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 623–632.

[19] H. Xiao, M. Huang, X. Zhu, From one point to a manifold: knowledge graph embedding for precise link prediction, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 1315–1321.
[20] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, in: International Conference on Learning Representations, 2018.
[21] M. Nickel, V. Tresp, H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 809–816.
[22] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33nd International Conference on Machine Learning, ICML, 2016, pp. 2071–2080.
[23] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, 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, pp. 601–610.
[24] 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 AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3060–3067.
[25] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, P.P. Talukdar, Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 3009–3016.
[26] Z. Xie, G. Zhou, J. Liu, X. Huang, ReInceptionE: Relation-aware inception network with joint local-global structural information for knowledge graph embedding, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5929–5939.
[27] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, 2020, arXiv preprint arXiv:2002.00388.
[28] M. Lin, Q. Chen, S. Yan, Network in network, 2013, arXiv preprint arXiv:1312.4400.
[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[31] Z. Sun, S. Vashishth, S. Sanyal, P. Talukdar, Y. Yang, A re-evaluation of knowledge graph completion methods, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5516–5522.
[32] 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, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 978–987.
[33] S. Zhang, Y. Tay, L. Yao, Q. Liu, Quaternion knowledge graph embeddings, in: Advances in Neural Information Processing Systems, 2019, pp. 2735–2745.
[34] T. Lacroix, N. Usunier, G. Obozinski, Canonical tensor decomposition for knowledge base completion, in: International Conference on Machine Learning, PMLR, 2018, pp. 2863–2872.
[35] D.Q. Nguyen, T. Vu, T.D. Nguyen, D.Q. Nguyen, D. Phung, A capsule network-based embedding model for knowledge graph completion and search personalization, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 2180–2189.
[36] D. Nathani, J. Chauhan, C. Sharma, M. Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4710–4723.
[37] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Trans. Knowl. Discov. Data (TKDD) 15 (2) (2021) 1–49.
[38] R. Kadlec, O. Bajgar, J. Kleindienst, Knowledge base completion: Baselines strike back, 2017, arXiv preprint arXiv:1705.10744.

**Fengyuan Lu** received the B.S. degree from the Department of Computer Science and Technology, South East University, Nanjing, China, in 2018. He is currently pursuing his Master degree with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His current research interests are in the areas of internet of things and knowledge graph embedding.

**Jie Zhou** received the B.S. degree from the Department of Computer Science and Technology, XiangTan University, Hunan, China, in 2019. He is currently pursuing his Master degree with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His current research interests are in the areas of deep learning and knowledge graph reasoning.

**Xinli Huang** received his Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently a full professor with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His recent research interests focus on the areas of Artificial Intelligence & Internet of Things, Smart Internetworking & Computing, and their applications in Smart Cities. He is a member of the IEEE.