

Research Article

Multipath Cross Graph Convolution for Knowledge Representation Learning

Luogeng Tian ^{1,2}, Bailong Yang,¹ Xinli Yin ³, Kai Kang ¹ and Jing Wu¹

¹High-Tech Institute of Xi'an, Xi'an 710025, China

²Information and Communication College, National University of Defense Technology, Xi'an, 710106, China

³National Innovation Institute of Defense Technology, Beijing 100071, China

Correspondence should be addressed to Luogeng Tian; tianlg82@163.com

Received 15 September 2021; Accepted 23 November 2021; Published 28 December 2021

Academic Editor: Thippa Reddy G

Copyright © 2021 Luogeng Tian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the past, most of the entity prediction methods based on embedding lacked the training of local core relationships, resulting in a deficiency in the end-to-end training. Aiming at this problem, we propose an end-to-end knowledge graph embedding representation method. It involves local graph convolution and global cross learning in this paper, which is called the TransC graph convolutional network (TransC-GCN). Firstly, multiple local semantic spaces are divided according to the largest neighbor. Secondly, a translation model is used to map the local entities and relationships into a cross vector, which serves as the input of GCN. Thirdly, through training and learning of local semantic relations, the best entities and strongest relations are found. The optimal entity relation combination ranking is obtained by evaluating the posterior loss function based on the mutual information entropy. Experiments show that this paper can obtain local entity feature information more accurately through the convolution operation of the lightweight convolutional neural network. Also, the maximum pooling operation helps to grasp the strong signal on the local feature, thereby avoiding the globally redundant feature. Compared with the mainstream triad prediction baseline model, the proposed algorithm can effectively reduce the computational complexity while achieving strong robustness. It also increases the inference accuracy of entities and relations by 8.1% and 4.4%, respectively. In short, this new method can not only effectively extract the local nodes and relationship features of the knowledge graph but also satisfy the requirements of multilayer penetration and relationship derivation of a knowledge graph.

1. Introduction

With the increasing construction of giant knowledge graphs, graph neural networks (GNNs), graph convolutional network (GCN) [1], and other neural networks that originally performed well on the graph appear to be incapable, and the calculation of adjacency matrix with full graph has become a problem. In many task scenarios, the entities in the map have close and important relationships with surrounding entities [2–4] and may have nothing to do with entities beyond a few steps. For example, TransE [5] series mostly only consider the direct relationship between entities. However, facts show that the rich and complex multistep relationships between entities in the knowledge graph are of great value for improving the quality of knowledge graph embedding. In a sense, the value of an entity lies in its interaction with other entities. This relationship can be quantitative or qualitative.

In other words, the same entity has relatively stable characteristic attributes in a fixed scene, and the relationship path is the necessary information supplement of the entity. The multilayer type of the entity mapped by its relationship is significant in knowledge representation learning or logical reasoning [6]. Moreover, various works have also been developed that support entity and relationship prediction [7–9]. For example, Hogan [10] replaced entities with canonical labels for solemnising existential nodes. Zhao et al. [11] proposed an effective method of using local relationships in entity type prediction.

A large number of knowledge graph cases show that a node often has a strong semantic relationship with a small number of adjacent nodes. It is not like GCN on “seeing flowers in the fog” and “finding needles in the sea” on the entire map, learning some valuable information from the giant knowledge map. Through the training and learning of

the model, we should enable it to accurately classify the types of entities and predict and judge the attributes of the classified entities. An example is shown in Figure 1 <LeBron Raymone James, player number, ? >.

The simple dot product of feature vectors and linear classification calculations can cause certain feature loss, which makes the entity classification and attribute prediction ineffective. As such, we propose a lightweight GCN for nonlinear cross learning of local knowledge graphs. Global iteration will certainly improve the coverage and accuracy, but this is often at the expense of the computational efficiency of the algorithm. At the same time, the knowledge graph structure or the baseline models of knowledge representations, such as TransE, TransR [12], and PTransE [13], considers the 1–3 step relationships and proves that the algorithm’s reasoning performance can be improved [14].

Given an example of such a set of triples as in Figure 2: <Titanic, leading actress, Kate Winslet>< Avatar, leading actress, Zoe Saldana>< Zoe Saldana, bra, Athletic undergarment><James Cameron, cooperative partner, LeBron Raymone James>< LeBron Raymone James, play for, Los Angeles Lakers>< LeBron Raymone James, another name, LBJ><LeBron Raymone James, another name, Zhan Huang><LeBron Raymone James, wife, Savannah James><LeBron Raymone James, player number, 23>.

To sum up, PTransE has a significant impact on the relational path embedding. It integrates multistep relational paths into knowledge representation learning, realizes information reasoning from the relational level, and improves the performance of knowledge graph completion. In terms of path selection, we proposed a resource allocation algorithm. Although this algorithm is feasible in quickly obtaining an effective relationship path, it is easy to cause resources to move closer to the entity that flows to the first step. If we select the first step of resource flow to the entity path, the first step of relationship path fitting will be generated. The main reason is that the co-occurrence relationship of global node features is ignored. Thus, a new local convolution global crossover named TransC-GCN is proposed in this study. It can not only effectively extract the local nodes and relationship features of the knowledge graph but also consider the needs of multilayer penetration and relationship derivation of the knowledge graph.

We are committed to embedding giant graphs and high-dimensional entities into low-dimensional entity type relationships. Figure 3 presents an example of learning and crossing through multiple local semantic spaces with similar classes, finally achieving two goals: (1) entity type judgment and classification and (2) entity name prediction.

Based on the research goals, we consider further adding local knowledge relationship path features on the basis of PTransE. At the same time, we combine multiple local crosses to realize the combination of local deep learning of relationship paths and global fusion representation.

Through the graph convolution learning of local entity relationship, not only can the hidden entities and relationship features in the local be discovered but also the local knowledge representation effect can be improved.

The continuous improvement of local knowledge representation ability will improve the overall knowledge learning performance and strengthen the knowledge reasoning ability.

The small-scale local graph convolution application can avoid the occurrence of global overfitting while solving the long calculation time.

In order to achieve the above goals, we need to address two challenges:

- (1) *Partial Division Problem*. If the part is too small, it will increase the amount of graph convolution calculation. On the contrary, the local features would be too rough. Therefore, partial division is the primary challenge.
- (2) *Cross Loss Function*. The iterative application of the local GCN improves the learning effect of local features by defining the loss function and constraining the optimization during the local crossover process. This is a key to ensuring the quality of the model.

Our contributions can be summarized as follows:

- (1) Using graph convolution combined with the out degree and in degree of the knowledge graph to iteratively calculate the local range, which can prevent the local division from being too large, we limit the local calculation to a certain threshold range related to the global graph structure.
- (2) The joint loss function is constructed through knowledge prior probability, posterior probability, and local cross entropy, which is calculated by normalization.

2. Related Works

2.1. GCN Full Graph Reasoning. Kipf et al. [15] introduced Spectral GCNs for semisupervised classification of spatial GCN graph structure data and applied convolution operations to calculate new feature vectors for each node with its neighborhood information. The fly in the ointment is that GCN needs to import the entire image to train the information and requires the training data to be unified with the verification data.

GCN combining features of nearby nodes is dependent on the structure of the graph, which limits the generalization ability of the trained model on other graph structures. Ermis et al. [16] believe that in a graph, predicting the link relationship between nodes can better study the entire graph network. For example, Wu et al. [17] used GCN to express the relationship between users and projects in the user-project structure diagram. Wang et al. [18] used GCN in KGs to improve the recommendation effect, while causing the hidden danger of overfitting and GCN performance degradation due to the lack of regularization. Wang et al. [19] realized the alignment of cross language knowledge graphs through graph convolutional networks.

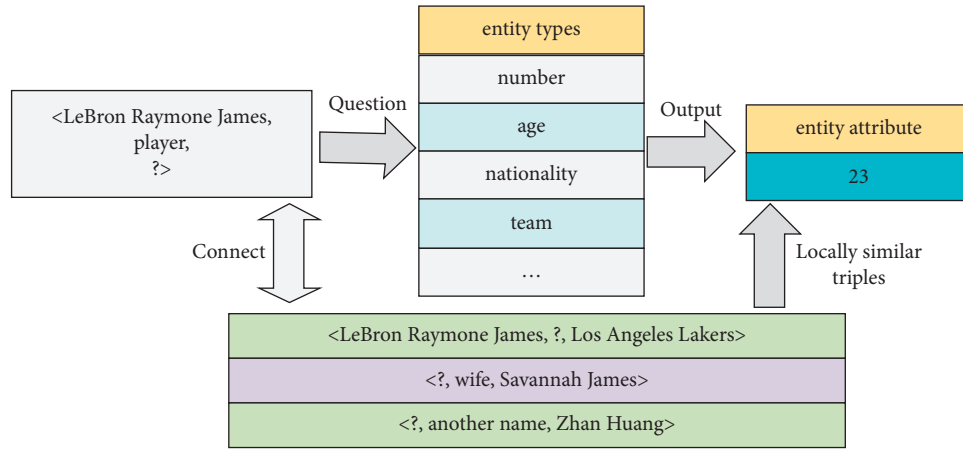


FIGURE 1: Node types and node predictions in the knowledge graph.

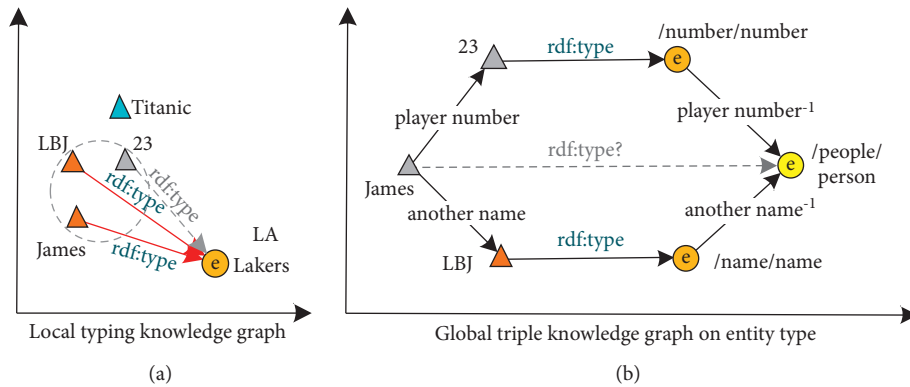


FIGURE 2: Local reasoning based on neighbor nodes and multihop reasoning in local graphs. (a) Mechanism 1. Infer potential nodes through neighbor nodes. (b) Mechanism 2. Through multihop key relationships in local knowledge graph.

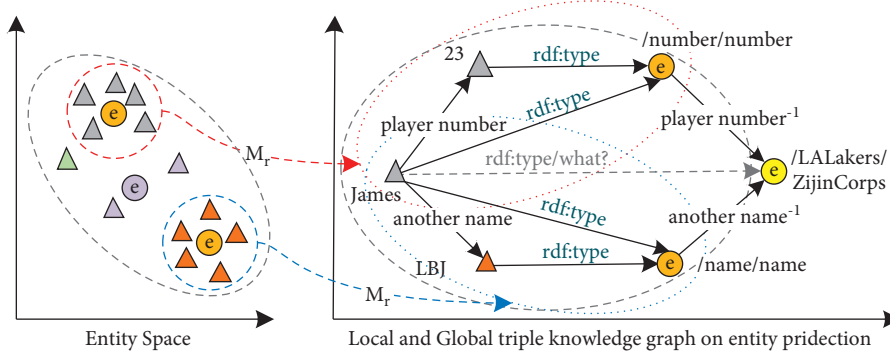


FIGURE 3: Schematic diagram of using multiple local knowledge graphs for cross inference and completion of triples.

2.2. *Neighbor Sampling Learning.* Graph Sample and Aggregate (GraphSAGE) is the most representative method in terms of uniform sampling of neighbor nodes and local node aggregation, as shown in Figure 4. By training the function of neighbors on the aggregated subgraph nodes, GCN is extended to inductive learning, thereby generalizing unknown nodes [18].

GAT [20] uses the attention distribution metric of neighbor nodes to weight and aggregate the local implicit information of the adjacency matrix. The local graph embedding representation of the central node is composed of

the feature representation of the central node and that of neighbor nodes. Through the splicing of node vectors, the feature representation of the center node can be iteratively updated, and then the feature representation of all nodes on the graph can be updated. In essence, GAT uses the feature aggregation function of the attention weight of neighbor nodes instead of the normalized function of GCN.

Unlike GCN, GAT allows implicitly assigning different importance to neighbors of the same node, while learning that attention is helpful for the interpretability of the model.

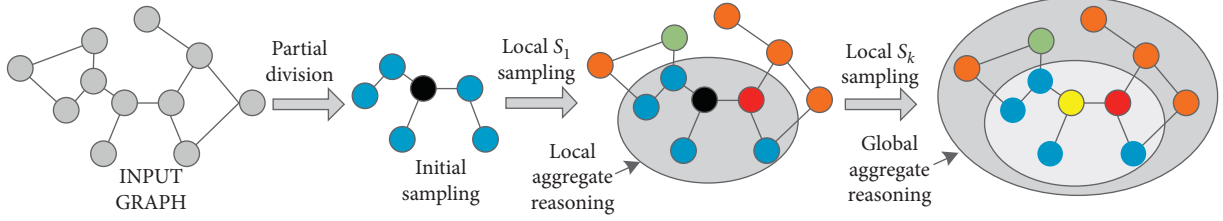


FIGURE 4: The local sampling and aggregation process of the graph.

The operation of GAT is point-by-point, and it is unnecessary to visit the global graph structure in advance. Therefore, it is suitable for inductive tasks. Important nodes in the graph and relations between nodes help to filter the noise between the neighbors of nodes and improve the interpretability of model results.

2.3. Embedding. A wide range of knowledge graph embedding techniques has been proposed. Based on the idea of TransE, the Trans(D,R) [21] defines the projection matrix M_r for the relationship r of each triplet $\langle h, r, t \rangle$ from the perspective of the relationship difference, and the head and tail entities are projected into the corresponding relational space. Then, TransE is used for translation. It is just that the head entity and the tail entity share M_r in the same triple, and there is no distinction between the head entity and the tail entity.

For example, $\langle \text{LeBron Raymone James, work for, Los Angeles Lakers} \rangle$, LeBron Raymone James is a person's name, and Los Angeles Lakers represents the collective. According to the above methods, we find that $\langle \text{James Cameron, director, Titanic} \rangle$ and $\langle \text{James Cameron, director, Avatar} \rangle$ are two triples with the same head entity and relationship. So, their tail entities are $\text{Titanic} \approx \text{Avatar}$. Obviously, this is incompatible with the fact. To solve this problem, TransD considers the difference between the head and tail entities. Similar to TransR, the head and tail entities are respectively projected into the relation r space; then, M_{rh} , h and M_{rt} are obtained. CrossE [22] uses a relational interaction matrix C to generate the interaction vector of the head entity and the relationship and then uses the vectors of these two interaction representations to predict the tail entity.

3. Algorithm Model

In this section, we propose an end-to-end knowledge map convolutional cross embedding representation method (TransC-GCN) as shown in Figure 5. Firstly, multiple local semantic spaces are divided according to the largest neighbor, and then a translation model is used to map the local entities and relationships into a cross vector, which is used as the input of GCN. Through training and learning of local semantic relations, the best entity and the strongest relationship are found. Finally, the optimal entity relationship combination output is evaluated through the posterior loss function based on the mutual information entropy.

The framework is mainly composed of 4 parts: (1) partial knowledge graph learning partitioning and embedding representation; (2) performing GCN coding on the partial graph and using the combined node relationship of the partial graph as input; (3) cross aggregating multiple partial knowledge graphs along with key relationships into reasoning nodes; and (4) sorting prediction nodes for multiple relationship paths.

3.1. Subgraph Division. There are often complex relationships between nodes in a local knowledge graph. Therefore, the minibatch method is used for reference to the set of local subgraphs, which is composed of central nodes. Besides, all nodes at the q -order subgraph of \mathcal{B} are presampled and stored in the traversal. Feature and label propagation are only propagated in the local map. So, we define the subgraph division as follows.

Definition 1. Given $\mathcal{E} = \{\mathcal{E}, \mathcal{R}\}$, assume $|\mathcal{E}| = n$, $|\mathcal{R}| = m$. Define its incidence matrix $A \in \mathbb{R}^{n \times m}$ as follows:

$$r_{ij} = \begin{cases} 1, & i \in E_j, \\ 0, & i \notin E_j. \end{cases} \quad (1)$$

Definition 2. The number of nodes is $n < \infty$. Use one $n \times n$ matrix to represent adjacency matrix \mathcal{Z} , which is defined as $A(\mathcal{Z}) = (a_{i,j})$, where n is the order of the graph. The set of $A(\mathcal{Z})$ features is called spectrum of the graph.

$$a_{i,j} = \begin{cases} 1, & (i, j) \in E, \\ 0, & \text{other.} \end{cases} \quad (2)$$

Definition 3. $\mathcal{E}_c = \{\mathcal{E}_c, \mathcal{R}_c\}$ represents a subgraph of the knowledge graph used for cross learning c , where \mathcal{E}_c is a collection of head and tail entities in the subgraph of C and \mathcal{R}_c is the collection of relations among all nodes. Neighbor nodes of $e_i^{(c)}$ are defined as $N(e_i^{(c)}) = \{D(e_i^{(c)}, e_j^{(c)}) \geq d^o\}$.

$$D(e_i^{(c)}, e_j^{(c)}) = \frac{1}{1 + \|\mathbf{e}_i^{(c)} - \mathbf{e}_j^{(c)}\|}, \quad (3)$$

$$d^o = \frac{1}{|C|} \sum_{j=1}^{|C|} \|\mathbf{e}_i^{(c)} + r^o - \mathbf{e}_j^{(c)}\|_{l_2}^2, \quad (4)$$

where $D(e_i^{(c)}, e_j^{(c)})$ denotes the Euclidean distance between the center node i to the neighbor node j and d^o is the

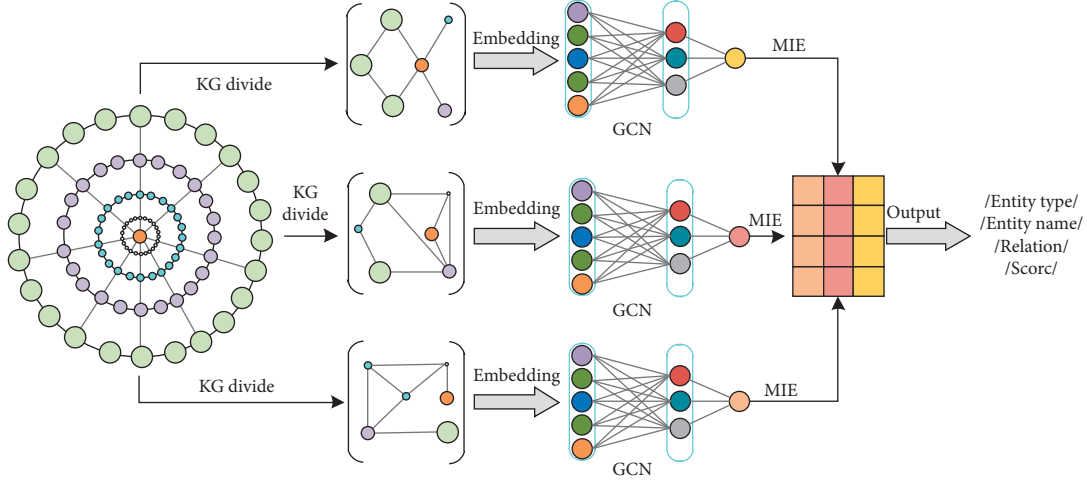


FIGURE 5: End-to-end knowledge graph convolution cross embedding representation frame.

relational path threshold of global graph. A local subgraph about the central node i is defined as $\mathcal{S}_i^{(c)}$. The schematic diagram of process is shown in Figure 6. Algorithm 1 shows the details of the calculation process.

3.2. Local Relation Graph Convolutional Coding. GCN can obtain the local information more accurately. Maximum pooling operation helps to grasp the strong signal on the local feature, thereby avoiding noise interference of the global redundant feature. Average pooling is used for neurons. The neighbor nodes connected to the central node are embedded as the input of a layer of neural network. Average pooling is used to eliminate the sparseness or overfitting problems of local nodes and relationship features. So, we use a layer of fully connected neural network and maximum pooling. Then, the output vector is multiplied by the feature vector of the central node to obtain a local embedding representation.

$$\mathbf{Y}_{out}(\mathcal{G}_c) = \sum_{m=1}^M \mathbf{D}_m^{-1/2} \hat{\mathbf{A}}_m \mathbf{D}_m^{-1/2} \mathbf{X} \mathbf{W}_m^{(\ell)}, \quad (5)$$

$$\hat{\mathbf{A}}_m = \mathbf{A}_m + \mathbf{I}_m. \quad (6)$$

The connection matrix $\hat{\mathbf{A}}_m$ is the label of m , $m \in \{1, 2, \dots, M\}$, and $\mathbf{D}_m^{-1/2}$ is the degree matrix corresponding to $\hat{\mathbf{A}}_m$.

$$\alpha_{ij}^{(c)} = \frac{\exp(\text{LeakReLU}((\mathbf{W}_h^{(\ell)} \mathbf{H}_c^{(\ell)} + b_s) + \epsilon^\circ))}{\sum_{k \in \mathcal{C}} \exp(\text{LeakReLU}((\mathbf{W}_h^{(\ell)} \mathbf{H}_c^{(\ell)} + b_s) + \epsilon^\circ))}, \quad (7)$$

where $\alpha_{ij}^{(c)} = (\alpha_{i1}^{(c)}, \alpha_{i2}^{(c)}, \dots, \alpha_{ic}^{(c)})$, $\mathbf{W}_h^{(\ell)}$ is the weight matrix between hidden layers, and b_s and ϵ° are the deviations. The nonlinear activation function σ is ReLU . Figure 7 shows the partial relationship diagram of the convolutional coding process.

3.3. Graph Relational Cross Matrix. In order to eliminate the overfitting of some important relationship nodes caused by the division of local graphs, multiple heads can be used to

calculate C subgraph branches in parallel. Then, all the subgraph branches can be defined by

$$\bar{\alpha}_{ij}^c = \frac{1}{|C|} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^c. \quad (8)$$

Local cross matrix:

$$\begin{aligned} \mathbf{C}_\alpha &= \mathbf{C}_\alpha \mathbf{C}_\alpha^T = [\bar{\alpha}_{ij}^1, \bar{\alpha}_{ij}^2, \dots, \bar{\alpha}_{ij}^m] [\bar{\alpha}_{ij}^1, \bar{\alpha}_{ij}^2, \dots, \bar{\alpha}_{ij}^m]^T \\ &= \begin{bmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mm} \end{bmatrix}. \end{aligned} \quad (9)$$

Definition 4. The graph cross matrix scaling factor is defined by the matrix mutual interference parameter [23], which measures the mutual interference parameters between different column vectors of local cross matrix. Specifically, the similarity relationship between the local graph structures can be found:

$$\rho = \max_{i \neq j} |c_i, c_j| = \max_{i \neq j} |c_i^H, c_j|. \quad (10)$$

From (9) and (10), the graph cross matrix of the fusion local graph structure is given by $\mathbf{C} = \rho \mathbf{C}_\alpha$.

3.4. Dynamic Node Prediction. The normalized attention coefficient is used to sum the weighted features as the preliminary output feature of each node:

$$\vec{e}'_i = \text{ReLU} \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{e}_j \right), \quad (11)$$

$$\vec{e}'_i = \text{ReLU} \left(\frac{1}{|C|} \sum_{c=1}^{|C|} \sum_{j \in \mathcal{N}_i} \bar{\alpha}_{ij}^c \mathbf{W}^{(\ell)} \vec{e}_j \right). \quad (12)$$

Local subgraph prediction node output:


```

(1)  $\mathcal{S}_i^{(c)} \leftarrow \mathcal{S}$ ;
(2) for  $c = |C|, \dots, 1$  do
(3)  $\mathcal{S}_i^{(c-1)} \leftarrow \mathcal{S}_i^{(c)}$ ;
(4) for  $e_{ij}^{(c)} \in \mathcal{S}_i^{(c)}$  do
(5)  $\mathcal{S}_i^{(c-1)} \leftarrow \mathcal{S}_i^{(c)} \cup \mathcal{N}_{\text{new}}(e_{ij}^{(c)})$ ;
(6) end
(7) end

```

ALGORITHM 1: Subgraph generation algorithm.

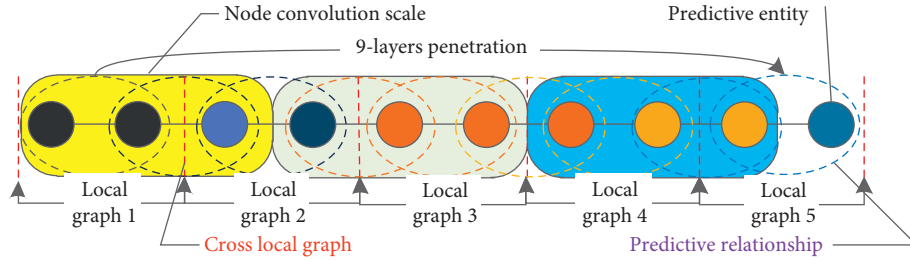


FIGURE 6: Predicting entities/relationships through multipart graph division and cross convolution.

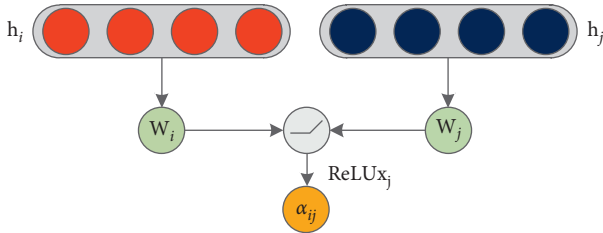


FIGURE 7: Partial relationship diagram of the convolutional coding process.

$$\hat{\mathcal{Y}}_{ci}(e_{ij}^{(c)}) = \sum_{ij \in \mathcal{N}(e_i^{(c)})} \frac{1}{\mathcal{S}_i^{(c)}(e_j^{(c)})} \mathbf{X}(e_j^{(c)}) \mathbf{W}^{(\ell)}(e_j^{(c)}), \quad (13)$$

where $\mathbf{X}(e_j^{(c)})$ is the characteristic of $e_j^{(c)}$; $\mathbf{W}^{(\ell)}$ is $e_j^{(c)}$ with GCN of shared weights on ℓ th layer; $\mathcal{S}_i^{(c)}$ is the normalized subset corresponding to all adjacent nodes $\mathcal{N}(e_i^{(c)})$ of $e_i^{(c)}$; and $\hat{\mathcal{Y}}_{ci}(e_{ij}^{(c)})$ is the target representation vector centered on $e_i^{(c)}$.

3.5. Loss Function

3.5.1. Triple Knowledge Embedding Loss Function. Cross correlation score of entities and relationships within triplet is used as the object of knowledge representation optimization. Here, the formal scoring function is given by

$$\mathcal{L}_{Emb} = - \sum_{(e_i^{(c)}, r_{ij}^{\circ}, e_j^{(c)})} \left[\log \Pr(e_i^{(c)} | r_{ij}^{\circ}, e_j^{(c)}) + \log \Pr(e_j^{(c)} | e_i^{(c)}, r_{ij}^{\circ}) + \log \Pr(r_{ij}^{\circ} | e_i^{(c)}, e_j^{(c)}) + \theta_1^{\circ} \right]. \quad (14)$$

3.5.2. Cross Convolution Loss Function. A graph convolution operator is inside the local knowledge graph, and multiple local graphs are cross fused based on the

information divergence. The local knowledge map features learned from the local convolution model are fed into the divergence fusion cross model. At the same time, we use gradient optimization algorithms of AdaGrad and stochastic gradient descent (SGD) to construct training modules. The estimated difference of partial subgraph output is defined as follows:

$$\begin{aligned} \min_N \mathcal{L}_{GCN} &= C_{JS D}(\mathcal{Y}_{ci}, \hat{\mathcal{Y}}_{ci}) \\ &= KL\left(\mathcal{Y}_{ci} \frac{\mathcal{Y}_{ci} + \hat{\mathcal{Y}}_{ci}}{2}\right) + KL\left(\hat{\mathcal{Y}}_{ci} \frac{\mathcal{Y}_{ci} + \hat{\mathcal{Y}}_{ci}}{2}\right) + \theta_2^{\circ} \\ &= \sum_{i \in A} \mathcal{Y}_{ci} \left[\log \frac{\mathcal{Y}_{ci}}{\hat{\mathcal{Y}}_{ci}} \right] + \sum_{i \in B} \hat{\mathcal{Y}}_{ci} \left[\log \frac{\hat{\mathcal{Y}}_{ci}}{\mathcal{Y}_{ci}} \right] + \theta_2^{\circ}. \end{aligned} \quad (15)$$

The loss function of global cross training error rate for the supervised training optimization model is shown as follows:

$$\begin{aligned} \min_C \mathcal{L}_{TransC} &= \min_C 1 - \sum_{i=1}^c \sum_{c \in C} \frac{\exp(-\|C^T \mathcal{Y}_c - C^T \hat{\mathcal{Y}}_{ci}\|_2^2)}{\sum_l \exp(-\|C^T \mathcal{Y}_c - C^T \hat{\mathcal{Y}}_{cl}\|_2^2)} + \theta_3^{\circ}, \\ \min_C \sum_{(\mathcal{Y}_c, \hat{\mathcal{Y}}_{ci}) \in \mathcal{M}} &\|\mathcal{Y}_c - \hat{\mathcal{Y}}_{ci}\|_M^2, \\ \text{s.t.} \sum_{(\mathcal{Y}_c, \hat{\mathcal{Y}}_{ci}) \in \mathcal{E}} &\|\mathcal{Y}_c - \hat{\mathcal{Y}}_{ci}\|_M^2 \geq 1, \quad M \geq 0. \end{aligned} \quad (16)$$

3.5.3. Joint Loss Function. In the training process, to effectively supervise the feature loss caused by local

TABLE 1: Comparison of characteristics of different datasets.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
FB15k [24]	14,951	1,345	483,142	50,000	59,071
WN18RR [29]	40,943	18	141,442	2,500	2,500
YAGO43k [30]	42,335	37	331,687	29,599	29,593
FB15kET	14,951	3,851	136,618	15,749	15,780
WN18R	40,501	10	110,341	2,100	2,100
YAGO43kET	41,723	45,182	375,853	42,739	42,750
FB15kTRT	3,851	1,345	2,015,338	—	—
YAGO43kTRT	45,128	37	1,727,707	—	—
WN18-R	6,501	5	90,341	—	—

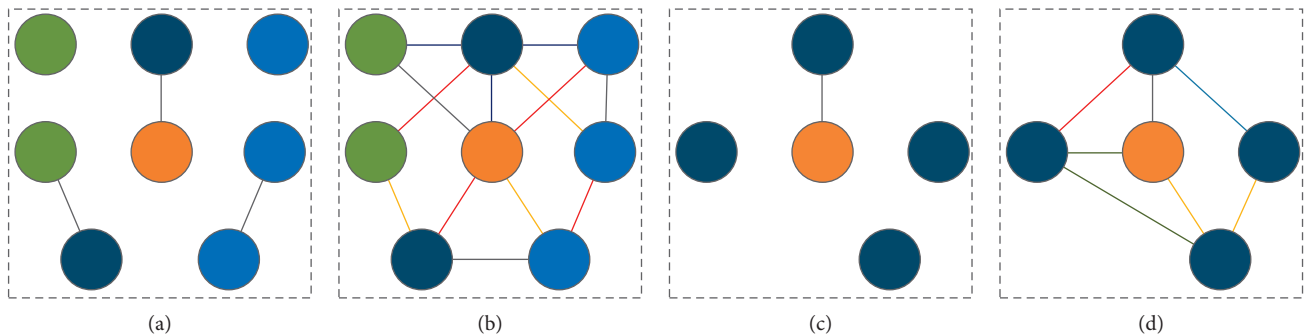


FIGURE 8: Sparse visualization of dataset entities and relationships. (a) Dense entities and lack of relationships of YAGO43k/WN18RR. (b) Dense entities and rich relationships of YAGO43kET. (c) Sparse entities and lack of relationships of WN18. (d) Sparse entities and rich relationships of FB15kTRT/FB15kET.

TABLE 2: The optimal parameter configuration of the corresponding datasets.

Parameters	Datasets		
	YAGO43kET	WN18RR	FB15kET
α	0.1	0.1	0.1
γ	0.6	0.5	0.4
q	200	150	250
ℓ	100	100	125
Batch size	4096	4096	4096
Epochs	300	300	300

convolution of the knowledge map and the partial filtering loss. When the divergence fusion crosses, the two tasks in the model are combined for training. This can improve the end-to-end training of the model. In summary, the joint loss function is written as

$$\mathcal{L}_{\text{TransC-GCN}} = \mathcal{L}_{\text{Emb}} + \gamma \mathcal{L}_{\text{GCN}} + (1 - \gamma) \mathcal{L}_{\text{TransC}} + \theta^\circ, \quad (17)$$

where $\gamma \in [0, 1]$ is a hyperparameter that adjusts the ratio of balanced coding and partial crossover and $\theta^\circ = \{\theta_1^\circ, \theta_2^\circ, \theta_3^\circ\}$ is the adjustment parameter.

4. Experiments

To further verify the effectiveness of TransC-GCN, we adopt the same experimental setting of Bordes et al. [24] in terms of entity prediction. In the evaluation index, the average ranking mean rank (MRR) and 10-hits rate (HITS@10) predicted by the entity are considered.

According to the experience of Shimaoka et al. [25], we use the pretrained word vector as the initialization and optimize the parameters with the optimizer Adam [26]. We use TransC-GCN to generate the representation vector of the triple. To avoid overfitting, we add dropout to the neurons of GCN and randomly inactivate neurons of the vector iteration.

Furthermore, we compare our model with multiple baseline models under different parameter settings on the two tasks of entity type classification and entity attribute prediction. Moreover, we compare it with the baseline model [27], where ConvKB(<https://github.com/daiquocnguyen/ConvKB>) and ConnectE-E2T + TRT(<https://github.com/Adam1679/ConnectE>) programs were run.

4.1. Datasets. In order to be more pertinent and comparative, we refer to the datasets extracted from the text relationship [28] as our research object. The characteristics of the three datasets are shown in Table 1.

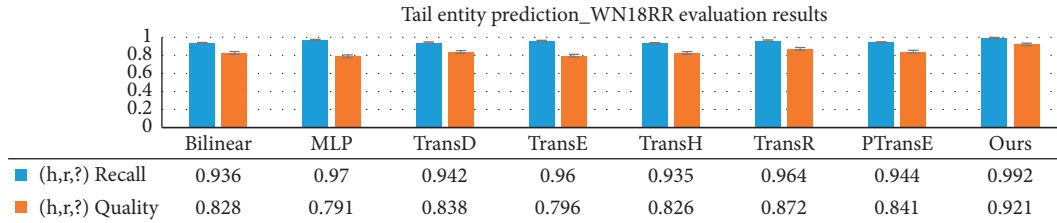


FIGURE 9: Comparison of baseline models for tail entity prediction.

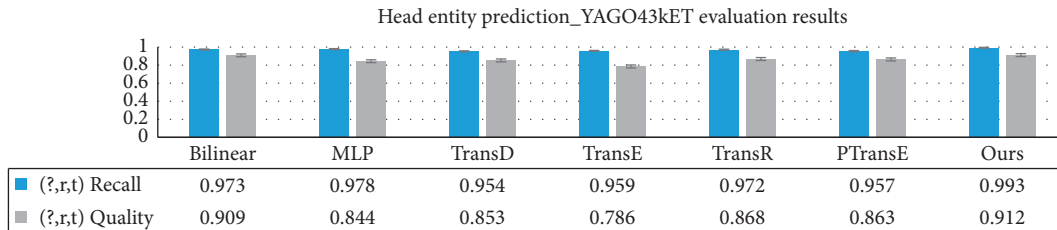


FIGURE 10: Comparison of baseline models for head entity prediction.

The visualization of the dataset intuitively shows that the number of entities and relationships and the degree of association have a greater impact on the local knowledge graph. Figure 8 shows that in the three datasets, the entities and relationships of YAGO43kET are dense and rich. On the contrary, WN18 has sparse entities and lacks relationships.

4.2. Evaluation Index. In order to verify TransC-GCN, we refer to two typical evaluation methods [31]. Formally, mean reciprocal rank (MRR) is defined as

$$MRR = \frac{1}{|N|} \sum_{i=1}^n \frac{1}{\text{Rank}_i}, \quad (18)$$

where N is the triplet number for the training dataset; Rank_i is the score averaged to the i th correct classification entity; and Hit rate is $H_{@K}$ ($K=1/3/10$), which means that in traversal training, the ability to obtain the correct triple entity prediction classification can be obtained once in K replacements.

4.3. Model Parameters. We conducted experiments on the training sets and the validation sets, as shown in Table 2.

We conducted cross explorations of different combinations in the setting of various parameters. According to the validation set effect of the corresponding dataset, the average ranking score will be the best.

4.4. Classification Prediction. Compared with tail entity and head entity prediction baseline models of the Bilinear, MLP and Trans series are shown in Table 3. WN18RRR with dense entities and poor relationships and YAGO43kET with dense entities and rich relationships are selected. For relationship

prediction verification, sparse entities and rich relationships of FB15kET are used.

4.4.1. Entity Prediction. As shown in Figures 9 and 10, TransC-GCN is better than the compared entity prediction models in both recall and quality indicators. The aggregation and intersection of key paths on the local knowledge graph can effectively improve the efficiency and quality of node prediction. Prediction index results of the head entity and tail entity of the triple completion are close to each other, which shows that the aggregation of critical paths based on local graphs has strong applicability for node prediction.

4.4.2. Relationship Prediction. Figure 11 shows that TransC-GCN also performs well in predicting the recall rate and quality of relationship evaluation. It not only considers the relationship path but also, more importantly, learns the characteristics of the knowledge graph through GCN, which can eliminate the random prediction of the model probability caused by the lack of triple entities or relationships to a certain extent. This provides richer necessary information for relationship path and entity prediction.

4.4.3. Robustness. As shown in Figure 12, on the same dataset, two different training optimization methods of TransC-GCN_SGD and TransC-GCN_AdaGrad are compared. We can observe from Table 4 and Figure 13 that the number of entities and the number of relationships have a significant impact on the convolution and crossover of local knowledge graphs. In addition, the comparison result proves that the maximum pooling is indeed better than the average pooling in solving feature redundancy. In a graph with sparse entities and lack of relationships, there is a serious data sparse problem, which leads to the long-tail distribution of entities and relationships. However, we take advantage of

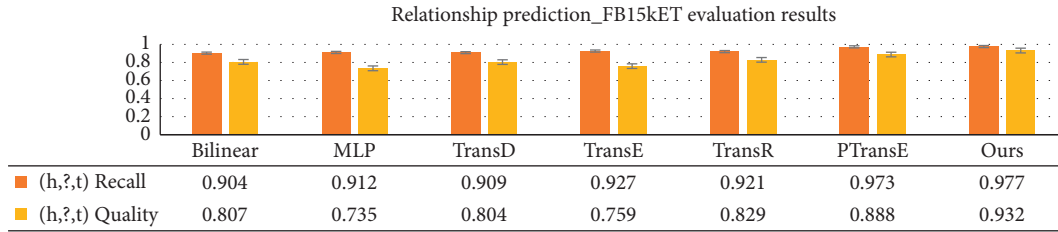


FIGURE 11: Comparison of relationship prediction of baseline models.

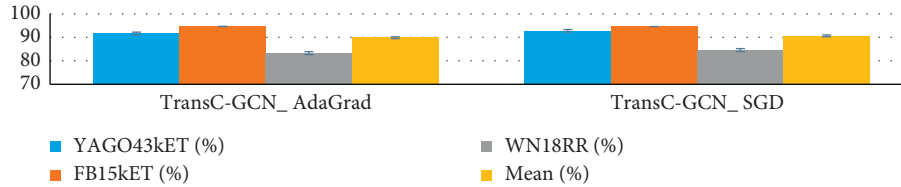


FIGURE 12: Robustness comparison of relationship prediction model on three datasets.

TABLE 4: Comparison of baseline models of entity type prediction accuracy.

Dataset	YAGO43kET (%)	FB15kET (%)	WN18RR (%)
TransE	85.36	92.15	75.04
TransE-ET	90.76	93.23	77.13
PTransE	88.53	92.79	78.25
ETE	90.82	94.01	79.31
ConnectE-(E2T+0)	91.78	94.45	81.07
ConnectE-(E2T + TRT)	92.33	94.49	81.22
TransC-GCN_ AdaGrad	91.63	94.61♣	83.27
TransC-GCN_ SGD	92.75♣	94.58	84.55♣

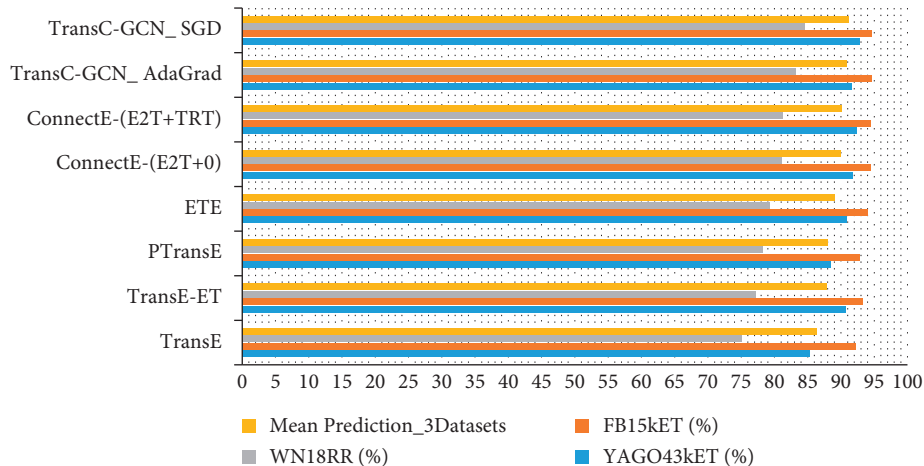


FIGURE 13: Comparison of baseline models of entity type prediction accuracy.

the critical path in the local graph, and the structure of the global graph is used.

4.5. Case Study. We have shown that our TransC-GCN can handle large-scale knowledge graph and entity-relationship

representation learning. In Figure 14, we provide an example of cross inference about local relationship paths. The core entity Savannah James is found through local cross learning. Two weak-strength relations (pay attention or like) are obtained by reasoning. We can find that Savannah James likes the 23 athletic undergarment bra of Zoe Saldana.

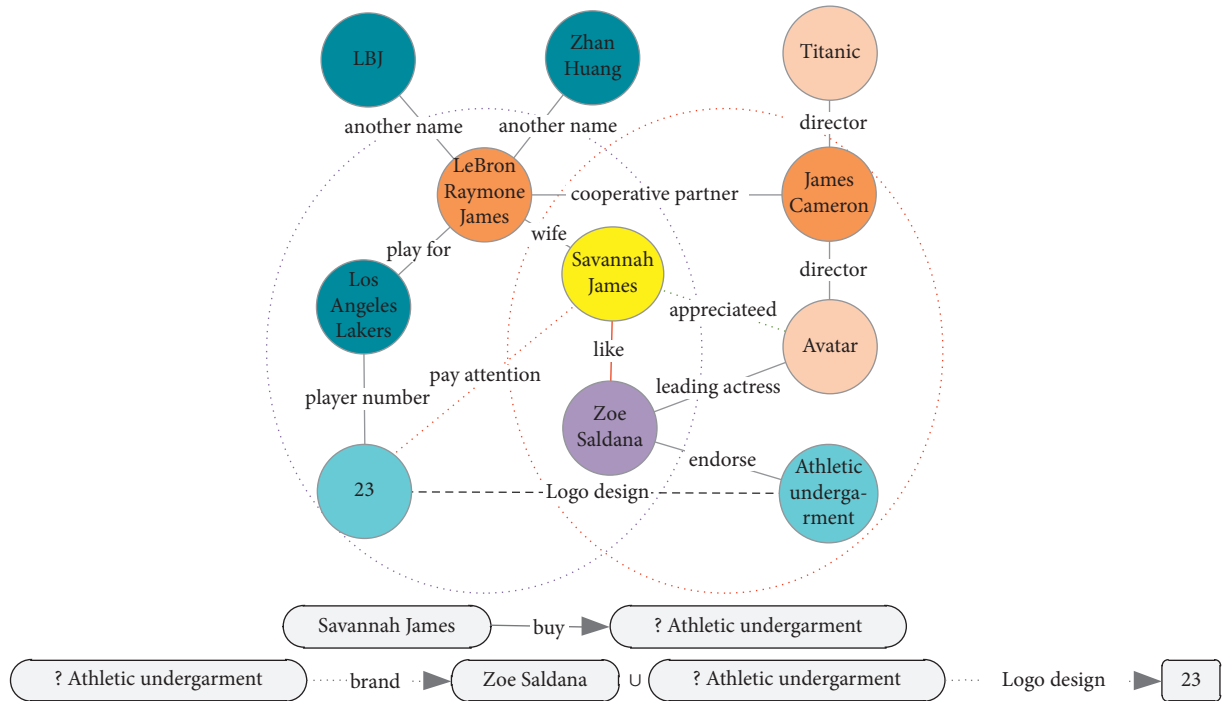


FIGURE 14: Entity node reasoning of user-commodity knowledge graph based on TransC-GCN.

Because Savannah James like Zoe Saldana, she appreciated Avatar. Maybe she likes the bra of Athletic undergarment, which is of the brand Zoe Saldana. Of course, Savannah James must pay more attention to the player number of 23, so she may be want a signed 23 athletic undergarment bra of Zoe Saldana.

5. Conclusion and Future Work

In this paper, we propose a TransC-GCN method based on local convolution and global crossover for knowledge graph completion. ①This is the first time that local GCN and TransC are combined for knowledge graph representation learning. ②TransC-GCN can not only divide the huge knowledge graph into several local knowledge graphs and use convolutional neural network coding for more intelligent and subtle local knowledge graph feature learning with strong semantic relations but also realize the volume of adjacent nodes and relationship features. ③Pooling and filtering data noise provide a new and efficient method for node relationship prediction and classification. ④TransC-GCN considers the information value of nearby local knowledge graphs. We propose a parallel method of cross fusion of local knowledge graphs based on divergence, which combines local knowledge graphs and global knowledge graphs more flexibly in representation learning. In a number of challenging baseline model test comparisons. TransC-GCN has excellent performance in entity reasoning accuracy and generalization ability and is lightweight. However, there are deficiencies in entity diversity learning:

- (1) When the gradient adopts AdaGrad to start training, the square of the accumulated gradient is found, which causes the effective learning rate of GCN to

decrease prematurely and excessively. After inactivating neurons with small gradients, ReLU is used as the activation function, resulting in loss of diversity.

- (2) Although experiments show that there is no significant difference between the maximum pooling and the average pooling in the results, there is still a loss of sparse entities in the test.
- (3) TransC-GCN is not efficient due to repeated calculations of neighbor nodes across subgraphs.

For future work, we will try to combine the pattern sequence ordering and the graph context to optimize TransC-GCN. Meanwhile, based on TransC-GCN, we intend to propose a new scene recommendation algorithm based on the combination of graph embedding and collaborative filtering.

Data Availability

Some or all data, models, and codes generated or used during the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Luogeng Tian was responsible for conceptualization, methodology, and original draft preparation. Bailong Yang was responsible for validation, formal analysis, and visualization. Xinli Yin was responsible for validation, formal

analysis, visualization, and software. Kai Kang was responsible for resources and review and editing. Jing Wu was responsible for supervision.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61501469).

References

- [1] T. N. Kipf, *Deep Learning with Graph-Structured Representations*, pp. 10–30, University of Amsterdam, Amsterdam, Netherlands, 2020.
- [2] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: a survey of approaches and applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.
- [3] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: problems, techniques, and applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.
- [4] N. Gupta, S. Singh, and D. Roth, “Entity linking via joint encoding of types, descriptions, and context,” in *Proceedings of the EMNLP 2017: Conference on Empirical Methods in Natural Language Processing*, pp. 2681–2690, Copenhagen, Denmark, September 2017.
- [5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 2787–2795, Lake Tahoe, NV, USA, July 2013.
- [6] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, “Relation classification via convolutional deep neural network,” in *Proceedings of the COLING*, pp. 23–29, Abingdon, England, June 2014.
- [7] T. Dettmers, M. Pasquale, S. Pontus, and S. Riedel, “Convolutional 2D knowledge graph embeddings,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1811–1818, San Francisco, CA, USA, February 2017.
- [8] B. Ding, Q. Wang, B. Wang, and L. Guo, “Improving knowledge graph embedding using simple constraints,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 110–121, Melbourne, Australia, July 2018.
- [9] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, “Learning attention-based embeddings for relation prediction in knowledge graphs,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Fortezza da Basso, Italy, 2019.
- [10] A. Hogan, “Canonical forms for isomorphic and equivalent RDF graphs: algorithms for leaning and labelling blank nodes,” in *Proceedings of the ACM Transactions on the Web*, no. 4, Las Vegas, NV, USA, June 2017.
- [11] Y. Zhao, A. X. Zhang, R. B. Xie, K. Liu, and X. J. Wang, “Connecting embeddings for knowledge graph entity typing,” in *Proceedings of the Association for Computational Linguistics*, pp. 6419–6428, Bangkok, Thailand, July 2020.
- [12] Y. K. Liu, S. Q. Shen, and Z. Y. Liu, “Neural relation extraction with selective attention over instances,” in *In Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 2124–2133, Berlin, Germany, August 2016.
- [13] Z. Y. Liu, X. Han, and M. S. Sun, *Knowledge Graph and Deep Learning*, pp. 32–39, Tsinghua University Press, Beijing, China, 2020.
- [14] M. Atef Mosa, “Predicting semantic categories in text based on knowledge graph combined with machine learning techniques,” *Applied Artificial Intelligence*, vol. 35, no. 12, pp. 933–951, 2021.
- [15] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.
- [16] B. Ermiş, E. Acar, and A. T. Cemgil, “Link prediction in heterogeneous data via generalized coupled tensor factorization,” in *In Proceedings of the DMKD*, pp. 203–236, Amsterdam, Netherlands, 2015.
- [17] Y. X. Wu, H. X. Liu, and Y. M. Yang, “Graph convolutional matrix completion for bipartite edge prediction,” in *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Seville, Spain, September, 2018.
- [18] H. W. Wang, M. Zhao, X. Xie, W. J. Li, and M. Y. Guo, “Knowledge graph convolutional networks for recommender systems,” *Proceedings of WWW*, vol. 3, 2019.
- [19] Z. Wang, Q. Lv, X. Lan, and Y. Zhang, “Cross-lingual knowledge graph alignment via graph convolutional networks,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 349–357, Brussels, Belgium, December 2018.
- [20] L. William, R. Y. Hamilton, and L. Jure, “Inductive representation learning on large graphs,” in *Proceedings of the Neural Information Processing Systems. Conference Series*, Long Beach, CA, USA, 2017.
- [21] Y. Wu, D. Bamman, and S. Russell, “Adversarial training for relation extraction,” in *Proceedings of the Encourages Submitting Software and data that is Described in the Paper as Supplementary Material*, pp. 1778–1783, Copenhagen, Denmark, September 2017.
- [22] W. Zhang, P. Bibek, and W. Zhang, “Abraham bernstein, chen H J. Interaction embeddings for prediction and explanation in knowledge graphs,” in *Proceedings of the ACM International Conference on Web Search and Data Mining*, Singapore, July 2019.
- [23] J. A. Tropp, “Greed is good: algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [24] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proceedings of the NeurIPS*, pp. 2787–2795, Montreal, Canada, December 2013.
- [25] S. Shimaoka, P. Stenetorp, and K. Inui, “Neural architectures for fine-grained entity type classification,” in *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 1271–1280, Vancouver, Canada, August 2017.
- [26] D. Kingma and J. Ba, “A method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, May 2015.
- [27] Y. Zhao, A. X. Zhang, R. B. Xie, K. Liu, and X. J. Wang, “Connecting embeddings for knowledge graph entity typing,” in *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 6419–6428, Bangkok, Thailand, July 2020.
- [28] S. Kiritchenko, S. Matwin, R. Nock, and A. F. Famili, “Learning and evaluation in the presence of class hierarchies: application to text categorization,” *Advances in Artificial Intelligence*, vol. 18, pp. 395–406, 2006.
- [29] T. Dettmers, M. Pasquale, S. Pontus, and S. Riedel, “Convolutional 2d knowledge graph embeddings,” in *Proceedings*

of the 32th AAAI Conference on Artificial Intelligence, pp. 1811–1818, New Orleans, LI, USA, 2018.

- [30] C. Moon, P. Jones, and N. F. Samatova, “Learning entity type embeddings for knowledge graph completion,” in *Proceedings of the Conference on Information and Knowledge Managemen*, pp. 2215–2218, Singapore, November 2017.
- [31] Z. Q. Sun, S. H. Vashishth, S. Y. Sanyal, P. Talukdar, and Y. M. Yang, “A re-evaluation of knowledge graph completion methods,” 2020, <https://arxiv.org/abs/1911.03903>.